Rheinische Friedrich-Wilhelms-Universität Bonn
Mathematisch-Naturwissenschaftliche Fakultät
Institut für Informatik III

Prof. Daniel Cremers

Newton Methods for Total Variation Minimization

Diplomarbeit
von

Mohamed Souiai

Danksagung

Hiermit möchte ich mich bei Dr. Thomas Pock für das interessante Thema, die engagierte Betreuung sowie die vielen Anregungen bedanken. Ich danke außerdem Dr. Bastian Goldlücke für die hiesige Betreuung und die interessante Vorlesung die mir einen tieferen Einblick in das Thema ermöglichste. Besonders möchte ich mich auch bei Prof. Daniel Cremers bedanken der mir stehts, bei Problemen und Fragen, mit Rat und Tat helfen konnte. Desweiteren geht mein Dank an meine Schwester sowie an Youssef Elmassoudi die mir beim korrekturlesen zur Seite standen. Zuletzt möchte ich mich bei meinen Eltern für die moralische und finanzielle Unterstztung, während dieser Arbeit, bedanken.

Abstract

The subject of this thesis is Newton's method applied to Total Variation (TV) minimization. We particularly consider the minimization of the Rudin-Osher-Fatemi (ROF) functional introduced in [57] which is one of the simplest variational models in computer vision. The straightforward application of the ROF model solves the perhaps most fundamental image restoration problem, namely image denoising, which is a preliminary step in many machine vision tasks e.g. object recognition. Additionally, the ROF model can be used to solve a two-phased segmentation model, as we will show in this work.

The most important aspect of the ROF model is that the TV regularization term allows for edges in the solution while disfavoring noise. Additionally, due to the convexity of the TV regularizing term, many tools from convex optimization can be used to tackle the ROF model. The challenge in solving the ROF model lies in the non-differentiability of the TV term as well as in its highly non-linear underlying PDE. This difficulty has pushed researchers to investigate alternative formulations [10, 13, 19] more well behaved and to devise new efficient optimization techniques for solving these formulations.

The most common method for minimizing the ROF model is the gradient descent method. Being a first-order method the gradient descent algorithm exhibits slow convergence, especially near the minimizer of the ROF model. However, since the advent of the GPU (graphics processing unit) programmability, it is known that first-order methods are well suited to be implemented on the highly parallel GPU architecture. This is due to their low complexity and their low memory requirement. This makes an acceleration, up to real time performance, of many variational problems possible.

Another promising approach to solve variational problems efficiently, is to utilize second-order methods and in particular Newton's method. Newton's method provides better convergence results and was considered in many works [19, 20, 35, 41, 51, 63]. However, its application to the different formulations of the ROF model is not straightforward as many challenges have to be faced. The objective of this thesis is to study these methods. Moreover, we provide a novel method that solves the dual formulation of a Huber regularized ROF model. This regularization avoids the so called staircasing effect which is a major drawback when using TV as a regularizing energy.

Organization of this Thesis

The thesis is organized as follows:

Chapter 1 gives an introduction to the Total Variation and its applications. The TV regularizer is motivated by comparing a selection of regularizers and by studying the underlying solution spaces. Then some properties of TV and its application in image denoising is presented. Furthermore we detail its employment as a geometrical regularizer in image segmentation. Finally, the equivalence of the ROF model with a two-phase image segmentation problem is shown.

Chapter 2 offers a survey of convex optimization techniques and provides a brief theoretical background. In particular , first-order and second-order methods are discussed where the advantage of the Newton method over first-order methods is made clear. This is done by a convergence analysis of the gradient descent method.

Chapter 3 presents minimization algorithms for solving the ROF model. Although the emphasis is on Newton methods, other interesting techniques such as optimal gradient methods, the graph-cut and the SOCP approach, are also presented. Moreover, a novel method that solves the dual formulation a Huber-regularized ROF model is proposed. This method is a Newton algorithm based on an active set strategy.

Chapter 4 begins by discussing implementation aspects of the Newton methods. As a second step the performance of our proposed method is showcased by demonstrating its denoising and staircasing reduction capabilities. Moreover, a comparison of the Newton algorithms with regard to the performance of minimizing the ROF denoising model, is carried out. Finally, segmentation by means of the ROF model, using our proposed Newton method, is demonstrated.

Chapter 5 concludes this work by giving a brief discussion which leads to some ideas for future work.

Contents

Chapter 1

Introduction: The Total Variation

The total variation model in Computer Vision was introduced in [57] in the context of image denoising. Total variation as a regularizer is favored in imaging problems because of its ability to recover edges in the image. This gives rise to many possibilities for its applications such as denoising, deblurring, inpainting, optical flow, stereo imaging and 3D reconstruction or segmentation. In this chapter we motivate the usage of total variation by studying a selection of regularizers, solution spaces and the properties of image representation in these spaces. Then some of the properties of the Total Variation semi norm are described. Moreover, we review two important applications of the Total Variation regularizer: image denoising and image segmentation. TV based denoising is discussed trough a study of the Rudin-Osher-Fatemi model. Finally, we detail convex formulations of the two-phase case of the Mumford-Shah segmentation model that involve TV as a convex geometric regularizer.

1.1 Image Models

The right choice for the model of the observed image and the recovered solution is crucial in a Computer Vision task. It allows us to give a proper mathematical description of these images. Additionally, image models impose a regularity by restricting the images to a certain function space. Regularity spaces express prior assumptions of an ideal image. In the following we study a suitable image space for the input and a selection of regularizing energies, before getting to two popular image models: The Sobolev and the bounded variation image model. These spaces share the same prior assumption on the image, namely the smoothness of its underlying signal.

1.1.1 The L^p Image Model

The Lebesgue space of p-integrable functions

$$L^p(\Omega) = \left\{ f : \int_\Omega |f(x)|^p dx < \infty \right\} \tag{1.1}$$

is a suitable function space for the input image since it contains all meaningful images and additionally the assumed noise. Here $\Omega \subset \mathbb{R}^2$ denotes the image domain, say a rectangle. The function $f : \Omega \to \mathbb{R}$ stands for the input image.
It is adequate to treat input images as points in the space $L^2(\Omega)$ since it has nice analytical properties like being a Hilbert Space and hence coinciding with its dual topological space. Unfortunately, the space $L^2(\Omega)$ is too big to distinguish between signal and noise when measuring the image information. Therefore it can not be considered as a solution space.

1.1.2 Regularity

Because the solution differs from the input image by a certain smoothness assumption, it is important to find a space that excludes unwanted oscillations from the image. The obvious regularity to control the level of noise is some measure of the image gradient:

$$\int_{\Omega} \phi(\nabla u)\, dx \tag{1.2}$$

with:

$$\nabla u = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right)^T \quad , \quad |\nabla u| = \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2}$$

The choice of the function ϕ dictates certain properties like isotropy/anisotropy and edge-preserving capabilities of the regularizer. Figure (1.1) illustrates 3 types of regularizing energies applied to a noisy image (a).

- In image (b) the function ϕ is the quadratic energy $\phi(p) = |p|^2$ whose involvement in (1.2) manages to remove the noise but clearly fails restoring discontinuities.
- In the case of figure (c) the utilized function is $\phi(p) = |p_1| + |p_2|$ hence the anisotropic (ℓ_1) norm which obviously cancels out the noise from the image while maintaining edges.
- The same goes for figure (d); a product of a regularization induced by the Euclidean norm as a function of the gradient i.e $\phi(p) = |p|_2$.

Note that in the case where $\phi(p) = |p_1| + |p_2|$ only particular edge directions are preferred in the process of reconstruction, namely the ones that are parallel to the sides of the so-called *Wolff shape* associated with $\phi(p)$:

$$W_\phi := \{y \in \mathbb{R}^2 : x \cdot y \leq \phi(x)\ \forall x \in \mathbb{R}^2\}. \tag{1.3}$$

The Wolff shape of $\phi(p) = |p_1| + |p_2|$ is the closed unite square. Whereas $\phi(p) = |p|_2$ has the unit circle as its underlying Wolff shape. Figure (1.2) shows the difference between the (ℓ_1) and (ℓ_2) reconstruction. One observes that in the first case the reconstruction has blocky structures while in the latter, edges in every direction are recovered. For a detailed study of anisotropic regularizers and the underlying Wolff shape, the reader is advised to look at the following reference [53].

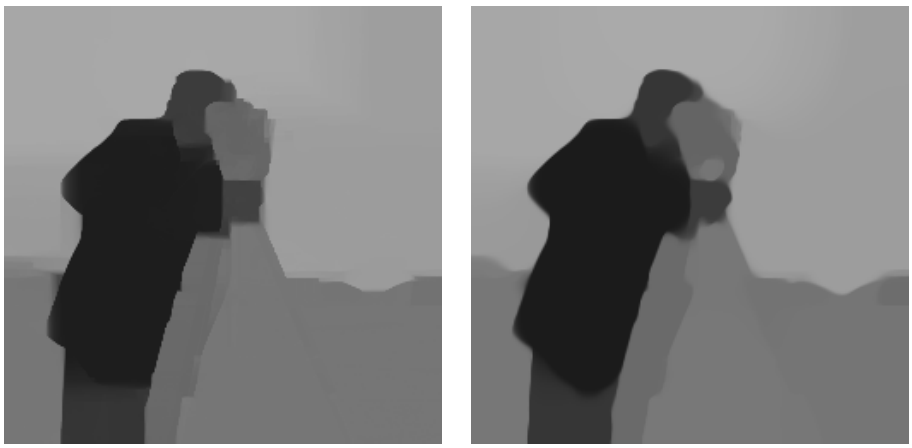
As highlighted above the measure $\int_\Omega = |\nabla u|^2\ dx$ produces smooth images without allowing for discontinuities. In fact the resulting functions belong to the so-called Sobolev space which is detailed next.

1.1.3 The Sobolev Image Model

Equation (1.2) suggests that the gradient of u have to be one-integrable. Therefore the Sobolev space $W^{1,1}(\Omega)$ is first image model that come to mind. The Sobolev space is the following collection of functions :

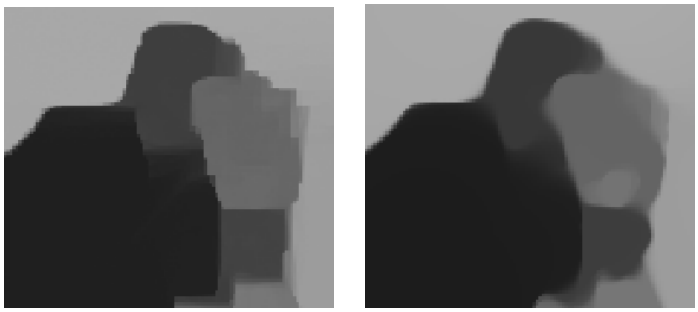
$$W^{1,1}(\Omega) = \left\{u \in L^1(\Omega);\quad Du \in L^1(\Omega)^2\right\} \tag{1.4}$$

(a) Noisy image (b) $\int_\Omega |\nabla u|^2$



(c) $\int_\Omega |\nabla u|_1$ (d) $\int_\Omega |\nabla u|_2$

Figure 1.1: Comparison of 3 regularizations of the noisy image (a). Image (b) is a result of a quadratic penalty. Image (c) results from an anisotropic regularization. Image (d) is isotropically regularized.



(a) (ℓ_1) (b) (ℓ_2)

Figure 1.2: (ℓ_1) and (ℓ_2) reconstructions.

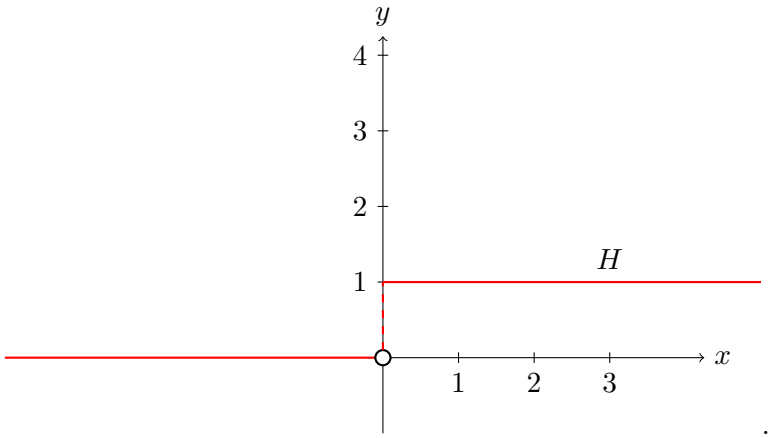


Figure 1.3: The Heaviside function.

hence it is the space of L^1 images with 1-integrable distributional partial derivatives up to the order of one. The distributional gradient coincides with the classical gradient, if u is differentiable, and is defined as the following linear functional:

$$< \nabla u, \varphi >= -\int_\Omega u \text{div } \varphi \, dx \qquad \forall \varphi \in D(\Omega)^2 \tag{1.5}$$

φ denotes a test function and lives in the infinitely differentiable function space $D(\Omega)^2 = C_0^\infty(\Omega)^2$ with compact support in Ω. Note that the Gauss-Green formula was applied in equation (1.5) and that no boundary conditions occur because of the compact support property of test functions. The image u is treated as a distribution and is a member of the dual space $D^*(\Omega)$ called the space of distributions. An interesting interpretation of test functions $\varphi \in C_0^\infty(\Omega; \mathbb{R}^2)$ is that they can be seen as linear sensors in the physical world which capture the image signal u e.g. CCD sensors or the human eye. For details see [17].

Still, the Sobolev space $W^{1,1}$ is too restrictive for solutions of the image processor because it does not allow the presence of edges in its functions. Consider for instance the one-dimensional piecewise constant function H:

$$H(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases} \tag{1.6}$$

which has a jump at the origin that can be interpreted as a one-dimensional edge. H is also referred to as the Heaviside step function which is the characteristic function of positive real numbers. The generalized derivative of H is the distribution H' satisfying:

$$\begin{aligned} < H', \varphi > &= (-1) \left\langle H, \frac{d\varphi}{dx} \right\rangle \\ &= -\int_{-\infty}^{\infty} H(x) \frac{d\varphi}{dx} \, dx \\ &= -\int_0^\infty \frac{d\varphi}{dx} \, dx \\ &= -[\varphi]_0^\infty \\ &= \varphi(0) - \varphi(\infty) = \varphi(0) =< \delta, \varphi > \end{aligned} \tag{1.7}$$

for all test functions φ. The calculation shows that the distributional derivative of H is the Dirac delta which is only defined in the distributional way:

$$< \delta, \varphi >= \varphi(0) \ \ \forall \varphi \in D(\Omega).$$

The Dirac delta is a singular distribution, because it cannot be generated by a function in L^p. Therefore H does not belong to $W^{1,p}$ for any $p \in [1, \infty]$. It is not difficult to generalize the above result to arbitrary piecewise constant functions in $\mathbb{R}^N$ with $N > 1$. An important example is the class of characteristic functions of closed sets which have derivatives that behave like the Dirac distribution. These derivatives are so-called Radon measures which are not valid gradients of a $W^{1,p}$ function.

1.1.4 The Space of Bounded Variation

Since the pioneering work of Rudin Osher and Fatemi [30] on edge-preserving image denoising, TV (Total Variation) methods became quite popular in the Computer Vision community. This results from the importance of discontinuities in images as they represent the most essential visual cue.

The BV (bounded variation) model assumes that an ideal image has bounded total variation. The Total Variation of a function $u \in L^1(\Omega)$ with $\Omega \subset \mathbb{R}^N$ is defined as follows [30]:

$$\int_\Omega |Du| = \sup \left\{ \int_\Omega u \mathrm{div}\ \varphi \, dx; \quad \varphi \in C_0^\infty(\Omega)^N, |\varphi|_\infty \leq 1 \right\} \tag{1.8}$$

Du is understood as the distributional derivative of u and the Lebesgue area element dx is omitted, indicating the more general Radon measure. The space of bounded variation is defined as:

$$BV(\Omega) = \left\{ u \in L^1(\Omega); \ \int_\Omega |Du| < \infty \right\} \tag{1.9}$$

and is a relaxation of the space $W^{1,1}$. Overall one has the following relations:

$$W^{1,1}(\Omega) \subseteq BV(\Omega) \subseteq L^2(\Omega) \tag{1.10}$$

The property (1.9) also holds true for piecewise constant functions since their distributional derivatives are bounded vector measures. Say a piecewise constant function (figure 1.4) $u : \Omega \to \mathbb{R}$ is partitioned in E and $\Omega \backslash E$ with:

$$u(x) = \begin{cases} 1 & \text{if } x \in E \\ 0 & \text{if } x \in \Omega \backslash E \end{cases} \tag{1.11}$$

The Total Variation of u is then given as:

$$\int_\Omega |Du| = \sup \left\{ \int_\Omega u \mathrm{div}\ \varphi \, dx; \quad \varphi \in C_0^\infty(\Omega)^N, |\varphi|_\infty \leq 1 \right\}$$

since u vanishes in $\Omega \backslash E$ and is 1 in E, one gets:

$$\sup \left\{ \int_E \mathrm{div}\ \varphi \, dx; \quad \varphi \in C_0^\infty(\Omega)^N, |\varphi|_\infty \leq 1 \right\}$$

The divergence theorem gives us the following integral over the boundary of Ω_1

$$\sup \left\{ \int_{\partial E} \varphi \cdot n \, ds; \quad \varphi \in C_0^\infty(\Omega)^N, |\varphi|_\infty \leq 1 \right\}$$

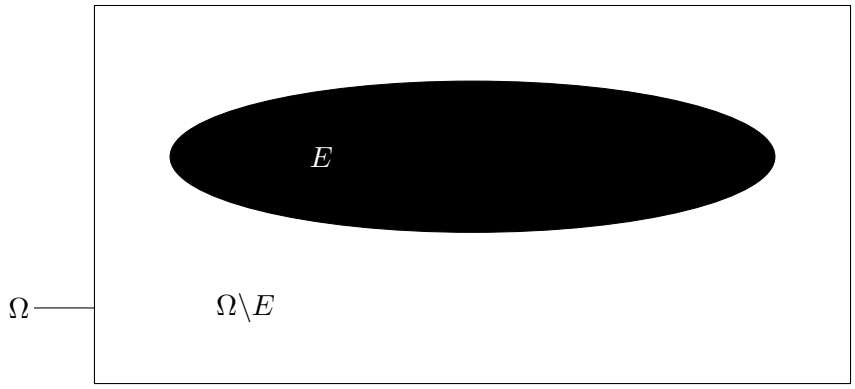


Figure 1.4: A two-phase piecewise constant function.

with n the outer unit normal to ∂E. The supremum is reached if we choose φ parallel to n and one gets:

$$\int_{\partial E} ds$$

which results in measuring the length of ∂E. As a consequence the piecewise constant function u is in $BV(\Omega)$ if the length of ∂E is finite. Sets with finite perimeter are also called BV sets. A more general connection between the TV-functional and the length of discontinuities will be given in the next section.

1.1.5 Properties of the Total Variation

Analytical Properties

The functional $J_{TV}(u) = \int_\Omega |Du|$ is a semi-norm on the space $BV(\Omega)$ and possesses the following properties:

- $J_{TV}(u)$ is convex.

  This follows because J_{TV} is a supremum of linear functions since the supremum operator preserves convexity.

- $J_{TV}(u)$ is lower semi-continuious i.e.

$$\liminf_{u \to u_0} \quad J_{TV}(u) \geq J_{TV}(u_0) \qquad \forall u_0 \in BV(\Omega)$$

  this property implies the closedness of $J_{TV}(u)$.

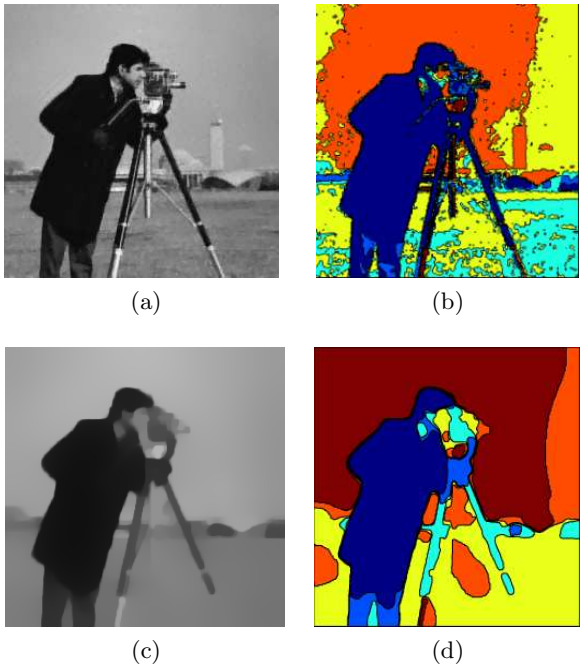
- $J_{TV}(u)$ is positive one-homogeneous i.e.

$$J_{TV}(\lambda u) = \lambda J_{TV}(u) \;\; \forall \lambda \geq 0.$$

- $J_{TV}(u)$ is proper; i.e it is not identically $-\infty$ or ∞.

  This is obvious because $J_{TV}(u)$ operates in the space of bounded variations.

The first and second property are important for the existence of a global minimizer for functionals containing $J_{TV}(u)$.



(a) (b)

(c) (d)

Figure 1.5: TV-minimization reduces oscillations in the isocontours. Image (c) is a TV regularized version of the original image (a) and has clearly smoother isocontours (d) (color coded) with reduced perimeter.

Geometric Properties

Originally formulated by Fleming and Rishel [26] the Co-Area formula gives a geometrical meaning to the TV semi-norm. For functions of bounded variation the following holds:

$$\int_{\Omega} |Du| = \int_{-\infty}^{\infty} Per(E_\gamma) d\gamma \tag{1.12}$$

E_γ represents the level domain:

$$E_\gamma = \{(x) \in \Omega |\ u(x) > \gamma\} \tag{1.13}$$

The effect of TV-regularization is illustrated in figure (1.5).

Suppose $u = \chi_D \subset BV(\Omega)$ is a characteristic function of a set $D \subset \Omega \subset \mathbb{R}^N$ and ∂D the boundary of D is smooth. In the last section it is shown that in this case the total variation measures the Euclidean length of D for $N = 2$ and the surface area for $N = 3$. It is this property that makes the TV regularizer also attractive for tasks where perimeter minimization is required like image segmentation and shape denoising. TV minimization is equivalent to smoothing the objects boundary in a binary function u, hence filtering out geometrical noise such as local ripples.

1.2 Image Denoising using Total Variation

Image denoising is perhaps the most fundamental image processing task and represents an important precursor to many computer vision tasks. Given a degraded image, the goal

is to recover the original signal without loosing the edge information in it. A very popular approach is realized by involving the total variation. This approach is well known as the ROF model and is detailed next.

1.2.1 The Rudin Osher Fatemi Model for Image Denoising

Image denoising based on the TV regularizer was first introduced in the work [57] of Rudin Osher and Fatemi. The ROF (Rudin Osher Fatemi) functional is one of the first models for denoising based on PDE's (partial differential equations) which exhibits edge preserving qualities. Originally the ROF model is posed as the following constrained optimization problem:

$$\inf \int_\Omega |\nabla u| \, dx \qquad s.t. \qquad \int_\Omega (u-f)^2 dx = \sigma^2 \tag{1.14}$$

Here, the original image is assumed to be corrupted with additive noise modeled by the Gaussian $N(0, \sigma^2)$. The operator $|\cdot|$ denotes the Euclidean norm. This constrained problem forces the optimization to go through functions u_k that are consistent with the noise level given in the constraint. Thus the minimization of the functional takes place in a non-convex set of functions which makes the whole problem non-convex although the objective is the convex TV functional.
Instead of solving the constrained optimization (1.14) problem, ROF and subsequent researchers proposed solving the following energy functional:

$$J_{ROF} = \int_\Omega |\nabla u| \, dx + \frac{1}{2\lambda} \int_\Omega (u-f)^2 dx. \tag{1.15}$$

This equation results from the relaxation of the constraint in (1.14) to

$$\int_\Omega (u-f)^2 dx \leq \sigma.$$

The parameter λ is a constant weight and does not depend on $x \in \Omega$. In fact λ acts as a Lagrange multiplier and expresses a tradeoff between TV regularity and the fitting term given by the least squares measure:

$$\int_\Omega (u-f)^2 dx.$$

This Lagrange multiplier should be adjusted to the apriori knowledge of the noise level ($\lambda \to 0$ for $\sigma \to 0$). The equivalence of the problems (1.14) and (1.15) was stated by Chambolle and Lions in [15].
The ROF-functional is a weighted sum of the convex TV-functional and the strictly convex quadratic fitting functional and is thus overall strictly convex. Consequently J_{ROF} admits a unique global solution. Figure (1.6) illustrates the denoising capability of the ROF model. One sees clearly that edges in the image are recovered.

The Euler-Lagrange PDE

The convexity of the ROF-functional facilitates stating the optimality conditions since one has to consider only the first variation. This leads us to the following Euler-Lagrange equation:

$$\frac{\partial J_{ROF}}{\partial u} = -\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \frac{1}{\lambda}(u-f) = 0 \tag{1.16}$$

(a) (b)

Figure 1.6: ROF minimization applied on a noisy image (a). The recovered image (b) has sharp discontinuities.

with the Neumann boundary condition $\frac{\partial u}{\partial n} = 0$ on the boundary $\partial\Omega$.

The authors in [57] also presented a time marching scheme for solving (1.16):

$$\begin{aligned} u_t &= -\frac{\partial J_{ROF}}{\partial u} = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) - \frac{1}{\lambda}(u - f) = 0 \\ u(0) &= f \end{aligned} \tag{1.17}$$

with the same boundary condition as in (1.16) and with the input image as the initial value.

The above descent direction u_t is not defined for the degenerated case $|\nabla u| = 0$. Such singularity is coped with by perturbing the term $|\nabla u|$ with a small positive constant ϵ i.e

$$|\nabla u|_\epsilon = \sqrt{|\nabla u| + \epsilon^2}. \tag{1.18}$$

Choosing a large value for ϵ, smears discontinuities and the functional loses its edge-preserving quality. A small ϵ causes numerical difficulties since the flux term $\nabla u / |\nabla u|$ could become arbitrarily large in flat regions. Either way the modified ROF model does not provide the same solutions as the original formulation. This fact along with the PDE (1.16) being highly nonlinear (due to the term $\nabla u \, |\nabla u|$) initiated a lot of research for establishing alternative formulations more faithful to the original ROF model. A review of these formulations will be given in chapter 3.

The ROF Functional as a Nonlinear Edge-preserving PDE

The gradient descent method (1.17) solves the diffusion PDE (1.16). The steady state of such this diffusion process is the solution of J_{TV}. The term $\frac{1}{|\nabla u|}$ corresponds here to a differential diffusion tensor and leads to a nonlinear diffusion filter. The nonlinearity of the PDE is caused by the dependency of the diffusion on the evolving image. Additonally, the filtering process is space-dependent because it is selective with regard to the differential diffusivity coefficient $|\nabla u|^{-1}$. The inhomogeneity of the diffusion PDE is responsible for going easy on edges as explained in the following:

- In homogenous regions $x \in \Omega$ where $\nabla u(x) \simeq 0$ the diffusion coefficient gets so small that the diffusion dominates.
- We also know from section (1.1.3) that the gradient $\nabla u(z)$ of egdes located in $z \in \Omega$ has the properties of the Dirac beam. This means that the magnitude of the gradient goes to infinity and the diffusion practically stops in $z \in \Omega$.

Consequently, and due to the TV-regularity, the ROF-model legalizes egdes in its solutions.

1.3 Image Segmentation using Total Variation

Image denoising and image segmentation are related because denoising algorithms try to spare important image features such as edges, which themselves are possible boundaries of objects. In this section, TV based convex formulations of non-convex segmentation functionals, are presented. Moreover, we state the equivalence between the ROF model and a two-phase segmentation problem. Priorly the concept of image segmentation, by means of another popular image model, is presented, namely the Mumford-Shah model.

1.3.1 Image Segmentation

While image Segmentation is usually a trivial task for the human vision it poses mathematically and computationally a great deal of challenge. The goal of image segmentation is to partition the observed noisy image in the domain Ω in to N meaningful disjoint regions:

$$\Omega = \Omega_1 \cup \Omega_1 \cup \cdots \cup \Omega_N \tag{1.19}$$

that hopefully belong to all the objects in the captured scene, including the background. Even though this definition of segmentation is intuitively clear at first, it is still complicated to characterize the definition of a meaningful partition, the number N of partitions and the notion of an "object". For example textures and shadows evoke discontinuities in images however they do not represent object boudnaries. This question is a subject of research in the works of Geman and Geman [29] and Mumford and Shah [47].

1.3.2 The Mumford-Shah Segmentation Model

One of the most known works on image segmentation is the image model introduced by Mumford and Shah in 1989 [47]. As an object-edge image model the MS (Mumford and Shah) -model assumes the image to consist of disjoint homogenous object patches $[u_k, \Omega_k]$ with $u_k \in H^1(\Omega_k)$ and regular boundaries $\partial\Omega_k$. Given an image $f : \Omega \subset \mathbb{R}^d \to [0; 1]$ with $d = 2, 3$ one searches for the set of discontinuities Γ and the piecewise smooth function u minimizing the following functional:

$$J_{MS}(u, \Gamma) = \frac{1}{2}\int_\Omega (u - f)^2 dx + \alpha \int_{\Omega\backslash\Gamma} |\nabla u|^2 dx + \lambda \mathcal{H}^{d-1}(\Gamma) \tag{1.20}$$

The MS functional consists of the following terms:

- The integral $\int_\Omega (u - f)^2 dx$ represents the fitting functional and penalizes differences to the input image.
- The term $\int_{\Omega\backslash\Gamma} |\nabla u|^2 dx$ is a Sobolev smoother hence u lies in $H^1(\Omega\backslash\Gamma)$ as the MS-model suggests. Thus, and opposed to the BV regularity condition, u is not allowed to have edges. This time such property is desired because the region $\Omega\backslash\Gamma$ is not supposed to have discontinuities by definition.

(a) (b)

Figure 1.7: An example of a piecewise smooth approximation using the MS model (a) is the original image f (b) is the piecewise smooth function u. These images are taken from [54].

- The Hausdorff measure $\mathcal{H}^{d-1}(\Gamma)$ is an extended notion of measure for even irregular surfaces or curves and for arbitrary d. In the case of $d = 2$ it measures the one-dimensional volume of Γ which is the length of all the curves in Γ. Involving the Hausdorff measure is necessary since a Lebesgue measure would fail to register edges which are Lebesgue zero sets. Minimizing the term $\mathcal{H}^{d-1}(\Gamma)$ results in eliminating fine scaled regions, such as isolated pixels, and disfavoring elongated structures. The parameter λ controls the length of the discontinuities.

An example of a Mumford-Shah approximated image is given in figure 1.7.

1.3.3 The Piecewise Constant Mumford-Shah Segmentation Model

The piecewise constant Mumford-Shah model rises from the asymptotic analysis $\alpha \to \infty$ of the above functional (1.20). Scaling α to infinity forces u to be absolute smooth in $\Omega \backslash \Gamma$ hence to have the constant intensity's $c_1 ...., c_N$ for the regions $\Omega_1 \ \cdots \ \Omega_N$. Since constant functions have a gradient of zero the smoothness constraint imposed on $\Omega \backslash \Gamma$ vanishes and the Mumford-Shah functional simplifies to:

$$J_{MS}(\{c_1 ...., c_N\}, \Gamma) = \frac{1}{2} \sum_{i=1}^{N} \int_{\Omega_i} (c_i - f)^2 \, dx + \lambda \mathcal{H}^{d-1}(\Gamma) \tag{1.21}$$

The fitting functional transforms into the L^2 penalty function of the distance from the input image partitions $f(\Omega_i)$ to the values c_i.
Differentiating with respect to c_i and stating the optimality condition:

$$\begin{aligned} \frac{\partial}{\partial c_i} J_{MS}(\{c_1, c_2 ...., c_N\}) &= \int_{\Omega_i} (c_i - f) \, dx = 0 \\ \Rightarrow c_i &= \frac{\int_{\Omega_i} f \, dx}{\int_{\Omega_i} 1 \, dx} \quad \forall i \in \{1; N\} \end{aligned} \tag{1.22}$$

(a) (b)

Figure 1.8: An example of a piecewise constant approximation using the MS model (a) the original image f (b) the piecewise constant function u with 15 regions. These images are taken from [2].

one can deduce that, given a partition $\Omega_1 \cdots \Omega_N$, the constant c_i equals the mean value of the input image intensities in the corresponding area Ω_i. Equation (1.22) gives an ideal approximation for the values c_i and one arrives at the following model for Mumford-Shah:

$$J_{MS}(\Gamma) = \frac{1}{2}\sum_{i=0}^{N}\int_{\Omega_i}(c_i - f)^2\,dx + \lambda\mathcal{H}^{d-1}(\Gamma) \text{ and } c_i = \frac{\int_{\Omega_i} f\,dx}{\int_{\Omega_i} 1\,dx} \tag{1.23}$$

which practically only hinges on the Contour Γ since c_i can be easily updated. The optimization of the objective (1.23) with respect to Γ is not a trivial task in particular when the curve/surface evolution needs to perform splitting or merging.

Note that the Mumford-Shah model is a non-convex optimization problem and is therefore hard to minimize. Recently Pock, Cremers, Chambolle and Bischof [54] presented a convex relaxation of the Mumford-Shah functional based on so-called functional lifting. An example of a solution for the piecewise constant Mumford-Shah functional is given in figure (1.8).

1.3.4 The Two-Phase Piecewise Constant Mumford-Shah Segmentation Model

Suppose u in the MS-model consists of only regions E and $\Omega\backslash E$. The function u is then assumed to have two values c_1 and c_2 i.e. :

$$u(x) = \begin{cases} c_1 & \text{if } x \in E \\ c_2 & \text{if } x \in \Omega\backslash E \end{cases} \tag{1.24}$$

with $E \subset \Omega$. The MS model then reads as:

$$J_{MS}(\{c_1....,c_N\},\Gamma) = \frac{1}{2}\int_E (c_1 - f)^2 + \frac{1}{2}\int_{\Omega\backslash E}(c_2 - f)^2\,dx + \lambda\mathcal{H}^{d-1}(\Gamma) \tag{1.25}$$

1.3.5 The Chan and Vese model

The authors Chan and Vese proposed in [16] a level set approach which handles the topological evolution of the curve effectively.

$$\begin{aligned} J_{CV}(c_1, c_2) &= \int_\Omega H(\phi(x))(f(x) - c_1)^2 \, dx + \int_\Omega (1 - H(\phi(x))(f(x) - c_2)^2 dx \\ &\quad + \lambda \, \mathcal{H}^1(\{\phi = 0\}). \end{aligned} \tag{1.26}$$

H is the following Heaviside function:

$$H(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases} \tag{1.27}$$

and acts as the characteristic function of the set E and respectively $1 - H(\phi)$ does so for $\Omega \backslash E$. The curves in this equation are represented as the zero level set of function ϕ therefore one can replace the Hausdorff measure with the length of the set $\{\phi = 0\}$ which measures the perimeter of E. The Chan and Vese functional is non-convex and the minimization gets stuck in local minima leading to undesirable segmentation results.

1.3.6 The Formulation of Nikolova et al.

In [18] Nikolova et al. showed that by fixing the constants c_1 and c_2 and by minimizing over the characteristic function u of the set E one obtains the following energy for the two-phase MS-model :

$$\min_{u \in BV(\Omega;\{1,0\})} \lambda J(u)_{TV} + \frac{1}{2} \int_\Omega u \left((c_1 - f)^2 - (c_2 - f)^2 \right) \, dx \tag{1.28}$$

The TV regularizer is incorporated to measure the perimeter of the set E. The above minimization is very difficult to carry out over the non-convex binary space $\{0; 1\}$. The authors, however, proposed a remedy for the difficult combinatorial problem (1.28) by relaxing the binary constraint to obtain the following convex formulation:

$$\min_{u \in BV(\Omega;[1,0])} \lambda J(u)_{TV} + \frac{1}{2} \int_\Omega u \left((c_1 - f)^2 - (c_2 - f)^2 \right) \, dx \tag{1.29}$$

which is almost the same optimization task, except u can take values in the interval $[0, 1]$. Thresholding u with a positive constant μ one gets the following representation for the set E:

$$E = \{u(x) > \mu\} \tag{1.30}$$

with $\mu \in [0, 1]$. Furthermore, it was shown in [18] that every value for the threshold yields to a global solution for the relaxed problem (1.29). This relaxation is known from integer linear programming and is referred to as LP-relaxation [1]. The global minimum is in general not unique and can depend on the chosen threshold. The segmentation results depend also on the chosen constants c_1 and c_2. Recently in [60] the authors also optimized over the parameters c_1 and c_2 and showed how to find optimal values for these variables by a simple branch and bound search over a single dimension.

Note that problem (1.29) is a constrained one. Nikolova et al. adress the problem by introducing the following additional penalty term:

$$\alpha \int_\Omega \nu(u(x)) \tag{1.31}$$

that makes sure u stays in the feasible set $[0,1]$. The function $\nu(s)$ looks as follows:

$$\nu(s) = \max\{0, 2|s - \frac{1}{2}| - 1\}$$

The resulting energy for solving the two-phase piecewise Mumford-Shah function is the following:

$$\lambda J_{TV}(u) + \frac{1}{2}\int_\Omega u\left((c_1 - f)^2 - (c_2 - f)^2\right) + \alpha \int_\Omega \nu(u(x))\, dx$$

1.3.7 Solving the Two-Phase MS Problem using the ROF Model

Next we describe a simpler way of solving the two-phase piecewise constant Mumford-Shah functional using only a thresholded solution of the ROF model and without any constraints involved. The following theorem will be helpful in doing so.

Theorem 1 *A function u^* is the solution of the generalized energy functional*

$$\min_u \lambda J_{TV}(u) + \int_\Omega G(x, u(x))\, dx \tag{1.32}$$

for a convex $G \in L^1$ which is smooth (C^1) in u, if and only if for any $s \in \mathbb{R}$ the binary function $\{u^* > s\}$ minimizes the following minimal surface problem:

$$\min_E \lambda Per(E) + \int_E \frac{\partial G(x,s)}{\partial s}\, dx \tag{1.33}$$

This theorem is stated by Chambolle and Darbon in [14]. The above equivalance is very important as it makes solving a whole family of minimal surface problems, in the form of (1.33), possible by minimizing the convex energy (1.32) and thresholding the solution. An interesting instance of theorem (1) is obtained by setting for G the following function:

$$G(x,s) = \frac{1}{2}\left(s - \left[(c_2 - f(x))^2 - (c_1 - f(x))^2\right]\right)^2 \tag{1.34}$$

This results in a shape optimization problem of the form:

$$\min_E \lambda Per(E) + \int_E \underbrace{s - \left[(c_2 - f(x))^2 - (c_1 - f(x))^2\right]}_{\frac{\partial G(x,s)}{\partial s}} \tag{1.35}$$

By setting $s = 0$ one obtains:

$$\min_E \lambda Per(E) + \int_E (c_1 - f(x))^2 - (c_2 - f(x))^2 \tag{1.36}$$

which is exactly the two-phase piecewise constant Mumford-Shah functional (1.28). Theorem (1) tells us that one can solve the two-phase piecewise constant Mumford Shah-functional by solving the following ROF problem:

$$\min_u \quad \lambda J_{TV}(u) + \frac{1}{2}\int_\Omega \left(u - \tilde{f}\right)^2 dx. \tag{1.37}$$

where the input data is replaced with the term

$$\tilde{f} = (c_2 - f)^2 - (c_1 - f)^2.$$

The binary function $\{u^* > 0\}$ is the global optimal solution to the optimization problem (1.36) and respectively to the two-phase piecewise constant Mumford and Shah functional (1.28).

Chapter 2

Convex Optimization

This chapter is devoted to numerical algorithms in convex optimization, some of their theoretical foundations and convergence properties. Mathematical optimization is the discipline of finding a solution to an optimization problem by minimizing a cost function. This chapter will give an overview of some of the techniques used for solving optimization problems. It starts off by describing the concept of an optimization problem and characterizing the existence, the uniqueness of its solutions and optimality conditions. Next an abstract concept for a minimization algorithm is given. Then the line search procedure is discussed. Furthermore first-order and second-order methods are presented. We motivate the usage of second-order methods by analyzing the convergence properties of the gradient descent algorithm which is a widely used first-order method. We then state the advantage of the Newton algorithm over first-order methods. Finally we look into the semi-smooth Newton algorithm, a method frequently used in this work.

2.1 The Optimization Problem

Let X be a normed space and f a real-valued function defined on the non empty closed convex subset $K \subset X$. The general idea of an optimization problem P is to find an element $u^* \in K$ such that:

$$f(u^*) \leq f(u) \quad \forall u \in K$$

Such an element can be described as:

$$f(u^*) = \inf_{u \in K} f(u).$$

if u^* exists we say f has a minimum at u^*.

If $K \neq X$ P is called a constrained optimization problem and on the contrary if $K = X$ then P is referred to as an unconstrained problem.

In this chapter we will consider only unconstrained convex optimization problems and the associated minimization techniques. The convexity of an optimization problem implies that every stationary point $\bar{u}$ satisfying:

$$\nabla f(\bar{u}) = 0 \tag{2.1}$$

is a global minimizer of f and a solution to problem P. A numerical algorithm solves the minimization problem P if it succeeds to find a critical point in the sense of (2.1).

Often the solution $\bar{u}$ is not given in a closed form and its determination is only possible by constructing a sequence $\{u_k\}$ such that:

$$u_k \to \bar{u}$$

where $\bar{u}$ is a stationary point and $k \to \infty$.

In the following we review the requirements for existence and uniqueness of the solution and how to recognize one.

2.1.1 Existence of the Solution

Assume $f : K \to \mathbb{R} \cup \{\infty\}$. The necessary conditions for the existence of a minimizer are the following two properties:

- *Lower semi-continuity*:

  For every sequence $u_k \to u$ we have:

  $$f(u) \leq \liminf_k f(u_k).$$

  Lower semi-continuity is equivalent with the closeness of a convex function hence all of its sublevel-sets are closed sets.

- *Coercivity*:

  A function f is called coercive if it is unbounded at the extremes of its domain. In fact f must satisfy:

  $$\lim_{||u|| \to \infty} f(x) = \infty.$$

Assume $\{u_k\}$ is a minimizing sequence of f so that:

$$\lim_{k \to \infty} f(u_k) = \inf_{v \in K} f(v),$$

then by the coercivity of f, the sequence $\{u_k\}$ must be bounded and according to Bolzano-Weierstrass must have a convergent subsequence $\{u_{k'}\}$. The closeness of f makes sure that the accumulation point $\bar{u}$ of $\{u_{k'}\}$ is in *dom* f and finally lower semi-continuity guarantees that $\bar{u}$ is the minimizer of f i.e

$$\inf_u f(u) \leq f(\bar{u}) \leq \liminf_k f(u_k) = \inf_u f(u).$$

In general Banach spaces the above existence result is not necessarily true since the underlying topology can be more complicated and boundedness does not imply compactness.

2.1.2 Uniqueness of the Solution

The convexity of the objective f means that every local minimum $\bar{u}$ is a global minimum. A convex function f is characterized by the following formula:

$$f(tu + (1-t)v) \leq tf(x) + (1-t)f(v) \tag{2.2}$$

for any $t \in [0, 1]$ and $u, v \in K$.
The uniqueness of such a point is still not a fact. The necessary condition for a minimum to be unique is the strict convexity of f. A function f is said to be strictly convex if strict inequality holds in (2.2) for distinct u and v and for any $t \in [0, 1]$.

2.1.3 Optimality Conditions

The famous mathematician Pierre de Fermat stated that if a real-valued differentiable function of one variable has an extremum at a point, its derivative at this point has to be zero. For differentiable convex functions f in $\mathbb{R}^n$ the statement of Fermat is true and the optimality condition for a global minimizer $\bar{u}$ reads as:

$$\nabla f(\bar{u}) = 0, \tag{2.3}$$

hence $\bar{u}$ must be a stationary point.

While condition (2.3) is sufficient for recognizing the global minimum of a convex differentiable functions in the discrete space $\mathbb{R}^n$, for infinite dimensional spaces the situation gets complicated and the optimality condition becomes more demanding.

The Gâteau Derivative

Dealing with convex functionals it suffices to consider the first variation, called the Gâteau variation. Let $J : X \to \mathbb{R}$ be a real-valued functional on a Banach space X, the mapping $GDJ : X \times X \to \mathbb{R}$

$$GDJ(u; v) := \lim_{t \to 0} \frac{J(u + tv) - J(u)}{t} \tag{2.4}$$

is called the Gâteau differential of $J(u)$ at a given u and is a set of directional derivatives towards all test functions $v \in X$. If the Gâteau differential happens to be linear and continuous in its second argument v and bounded, one can write instead $\langle DJ(u), v\rangle$ for the directional derivative. In this case $DJ(u)$ denotes the Fréchet derivative and the mapping $u \to DJ(u)$ is called the gradient of F which is denoted by ∇F in most literature. Note that ∇ maps into the topological dual X^* of X hence $DJ(u) \in X^*$, and that $\langle \cdot, \cdot \rangle$ stands for the duality pairing $\langle \cdot, \cdot \rangle_{X,X^*}$. For the sake of simplicity we assume $X = L^2(\Omega)$ hence $X^* = X$ and for $f, g \in X$:

$$\langle f, g\rangle_{X,X^*} = \langle f, g\rangle_{L^2} := \int_\Omega fg \, dx$$

The Euler-Lagrange Equation

For a Gâteau differentiable functional J the following optimality condition

$$GDJ(\bar{u}; v) \geq 0 \quad \forall v \in X$$

must hold at a stationary point $\bar{u}$. Moreover, if $J(u)$ is Fréchet differentiable, which means there exists a linear functional $DJ(u) \in X^*$ that describes the local variation of J at u in the following way:

$$\lim_{||v|| \to 0} \frac{||J(u + v) - J(u) - DJ(u)v||}{||v||} = 0 \quad \forall v \in X$$

where v is again assumed to be a test function, then the following holds at an optimal point $\bar{u}$:

$$\langle DJ(\bar{u}), v\rangle = 0 \quad \forall v \in X$$

This weak form of optimality condition suggests that

$$DJ(\bar{u}) = 0. \tag{2.5}$$

Equation (2.5) is called *Euler's* equation. Most of the times one is particularly interested in the representation of $DJ(\bar{u})$ in X^*. Such formulation can be tackled to solve (2.5) directly or used as an aid to determine the step direction in an optimization scheme, as will be shown in the optimization methods described in this work. Consider e.g the functional of the following integral form:

$$J(u) = \int_\Omega f(x, u, u')dx \qquad \left(u'(x) = \frac{du}{dx}\right). \tag{2.6}$$

This is a common formulation in the calculus of variation and a special case of the optimization problem P. The fact that the directional derivative at u can be written as the linear functional $DJ(u)$ applied on the direction (test function) v i.e.

$$\lim_{t\to 0} \frac{J(u+tv) - J(u)}{t} = \langle DJ(u), v\rangle = \int_\Omega DJ(u)v \; dx$$

suggests transforming the left hand side of the equation to the right hand side formulation $\int_\Omega DJ(u)v$ and extract $DJ(u)$ afterwards. We do so in the following for functional (2.6):

$$\lim_{t\to 0} \frac{J(u+tv) - J(u)}{t} = \lim_{t\to 0} \frac{1}{t} \int_\Omega f(x, u+tv, u'+tv') - f(x, u, u') \; dx \tag{2.7}$$

The first-order Taylor expansion of the first integrand gives us:

$$f(x, u+tv, u'+tv') = f(x, u, u') + t\left(v\frac{\partial f}{\partial u} + v'\frac{\partial f}{u'}\right) + o(t^2) \tag{2.8}$$

substituting (2.8) in (2.7) and taking the limit leads to:

$$\int_\Omega DJ(u)v \; dx = \int_\Omega \left(v\frac{\partial f}{\partial u} + v'\frac{\partial f}{u'}\right) \; dx$$

after integrating the second term by parts and knowing that test functions v vanishes on the boundary of Ω ($\Rightarrow$no boundary integral) one can recover $DJ(u)$ i.e.

$$\int_\Omega \underbrace{\left(\frac{\partial f}{\partial u} - \frac{d}{dx}\left(\frac{\partial f}{u'}\right)\right)}_{DJ(u)} v \; dx$$

$$DJ(u) = \left(\frac{\partial f}{\partial u} - \frac{d}{dx}\left(\frac{\partial f}{u'}\right)\right).$$

If we again demand the Eulers equation to hold at an optimal point:

$$DJ(\bar{u}) = \left(\frac{\partial f}{\partial \bar{u}} - \frac{d}{dx}\left(\frac{\partial f}{\bar{u}'}\right)\right) = 0 \tag{2.9}$$

we end up with a strong optimality condition for energy functionals of the type (2.6), well known as the *Euler-Lagrange equation.*

Optimality Conditions for Non-differentiable Functionals

Often the functional $J(u)$ is not Féchet differentiable. In this case the notion of the subdifferential arises. A subdifferential of J at u is the following set valued function:

$$\partial J(u) := \{p \in X^* | J(u) + \langle p, v - u \rangle \leq J(v)\} \ \forall v \in X. \tag{2.10}$$

The elements p are so-called subgradients and the affine function

$$H(v) = J(u) + \langle p, v - u \rangle$$

is a support hyperplane of the epigraph of J at $(u, J(u))$. In the one-dimensional case $H(v)$ is simply a tangent of the epigraph $(u, J(u))$ and the subgradient represents its slope. This is illustrated in figure (2.1). The generalized requirement for a u^* to minimize

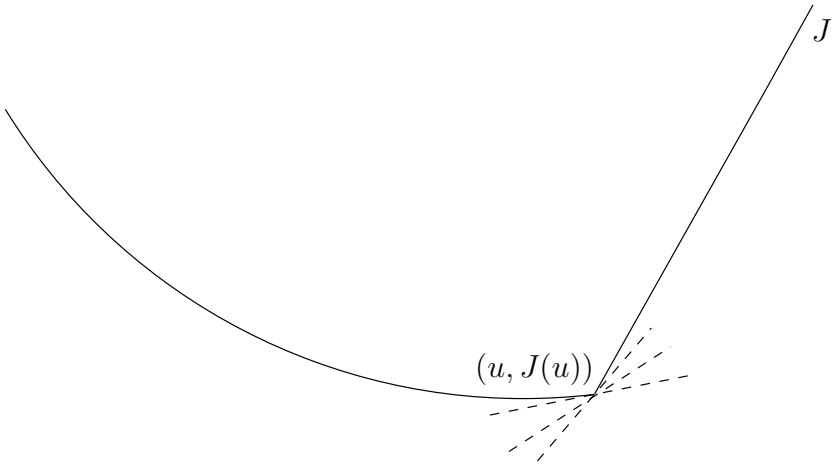


Figure 2.1: The Subdifferential of J at u is a set of slopes corresponding to tangents (dashed lines) that support the epigraph at $(u, J(u))$.

a non-differentiable functional is the following:

$$0 \in \partial J(u^*) \tag{2.11}$$

An example of a non Fréchet-differentiable function is the absolute function $g(u) = |u|$. $g(u)$ pocesses the sub-differential $\partial g(0) = [-1, 1]$ at the non-differentiable point 0. Since $0 \in [-1, 1]$, function $g(u)$ attains its optimal point at 0. For a detailed treatment of subdifferential calculus the reference [36] is recommended.

2.2 General Structure of Optimization Schemes

An optimization scheme is a set of rules to solve the above problem P iteratively. The main task of such an algorithm is to construct a minimizing sequence u_k that converges to the desired stationary point u^* for $k \to \infty$. Of course numerically k should be finite and it has to be defined a stopping criterion by measuring the proximity to the solution. That is the user decides whether a point u_k is "good enough" or not by specifying a minimum accuracy. The most natural stopping criterion is to measure the magnitude of the gradient and to stop if it falls below a specified constant δ i.e if:

$$||\nabla f(u_k)|| \leq \delta \tag{2.12}$$

u_k is the sought solution with sufficient accuracy.

This chapter will mainly survey techniques that force the objective f to decrease in every iteration:

$$f(u_{k+1}) < f(u_k) \tag{2.13}$$

this leads to a scheme called the *descend algorithm*.

A typical descend algorithm is designed to perform the following two tasks in every iteration:

- **Direction finding**:
  This stage is dedicated to find a suitable direction d_k along one searches for the next iterate u_k with a lower function value $f(u_k)$.

- **Line-search**:
  The second procedure consists of solving the following one-dimensional minimization problem:

  $$\min_{t>0} f(u_k + td_k) \tag{2.14}$$

  which amounts to finding a step length t such that the next iterate u_{k+1} minimizes the function along d_k and is updated by:

  $$u_{k+1} = u_k + td_k. \tag{2.15}$$

The descent Algorithm is summarized in algorithm 1 . The following section will demon-

Algorithm 1 The general descent algorithm.

1. Choose an initial point $u_0 \in domf$ and set k=0.
2. Find a d_k for which:

   $$\exists t_k > 0 \quad \text{s.t} \quad f(u_k + t_k d_k) < f(u_k) \tag{2.16}$$

3. Choose a step size t_k.
4. Stop or goto (2) and set $u_{k+1} = u_k + t_k d_k$ and $k = k + 1$.

strate the line search procedure.

2.3 Line Search

All optimization algorithms compute a search direction d_k and decide to move along this direction. Since the full step length does not always give a decrease in the objective function an appropriate choice for the step length t has to be found in every iteration. This is done via *line search*, an optimization along the ray of the direction d_k. Line search is performed to ensure global convergence, hence a convergence independent of the initial point u_0.

2.3.1 Exact Line Search

Ideally the step length t is chosen as the optimal point of the following univariate optimization problem:

$$\min_{t>0} f(u_k + td_k) \tag{2.17}$$

For a step length t computed as in (2.17) the procedure is called exact line search. Exact line search methods are particularly useful when the minimizer of (2.17) t^* can be found analytically or the cost of the one-dimensional minimization problem is not too high since the optimization is performed every single iteration. These circumstances are seldom a fact and for most purposes a less expansive procedure is required.

2.3.2 Backtracking Line Search

In practice line search techniques do not solve the optimization problem (2.17) exactly, but rather try out several step lengths and choose a candidate that causes an adequate reduction in the objective function. A very effective technique that follows this strategy is known as backtracking line search and is given in algorithm 2.

Algorithm 2 Backtracking line search.

Given the search direction d_k for f at u_k, $\alpha \in (0, 0.5)$, $\beta \in (0, 1)$ $t := 1$.

Repeat until $f(u_k + td_k) \leq f(u_k) + \alpha t \nabla f(u_k)^T d_k$

$$t := \beta t;$$

Thereby the *sufficient decrease* caused by the step length t is measured by the inequality:

$$f(u_k + td_k) \leq f(u_k) + \alpha t \nabla f(u_k)^T d_k \tag{2.18}$$

The directional derivative $\nabla f(u_k)^T d_k$ measures the decrease of f in u_k. By convexity of f the extrapolation $t \to f(u_k) + t\nabla f(u_k)^T d_k$ is a lower bound on $t \to f(u_k + td_k)$ and is never reached for any $t > 0$. This is why one has to degrade the expected reduction of f by a constant α. Figure (2.2) shows that the extrapolation $f(u_k) + t\nabla f(u_k)^T d_k$ touches f_{u_k} in u_k and lies strictly under the graph of $t \to f(u_k + td_k)$ while the weighted extrapolation $f(u_k) + \alpha t \nabla f(u_k)^T d_k$ intersects its graph for a $t > 0$. The condition (2.18) states that the step length t should give a better reduction than the weighted extrapolation $f(u_k) + \alpha t \nabla f(u_k)^T d_k$. Inequality (2.18) is referred to as the *Armijo condition*.
The backtracking procedure initializes t with the full step length of one and reduces it by β until the Armijo condition holds.
Even if the backtracking algorithm does not find the exact minimum t^* as in exact line search, it is shown (see e.g. [58]) that in practice this variant of line search does not change the order of convergence and should therefore be favored over an expansive search for the exact minimizer.

2.4 Defining the Direction

The most important property for the search step d_k is to decrease the objective f at every iteration:

$$f(u_{k+1}) < f(u_k) \tag{2.19}$$

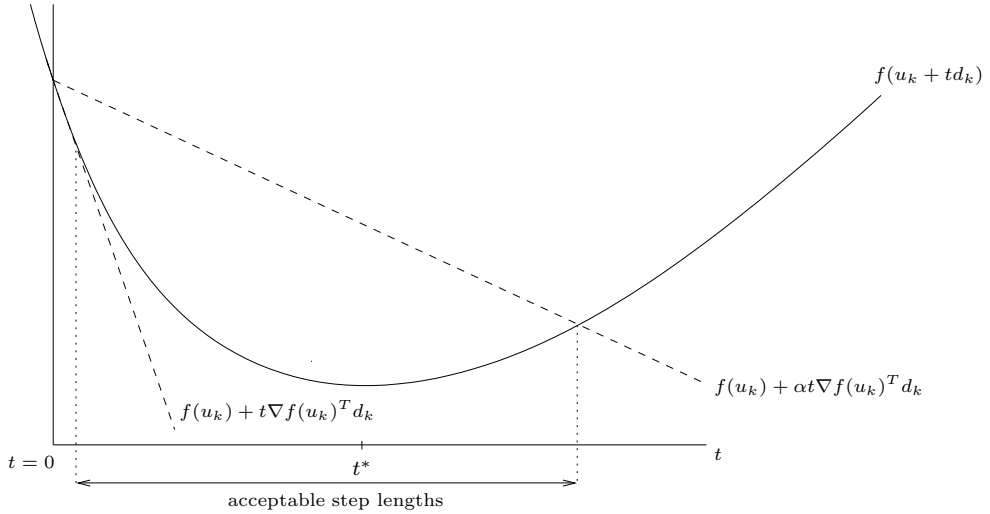


Figure 2.2: **The backtracking line search**

The convexity property of the objective function implies:

$$\langle \nabla f(u_k), (v - u_k) \rangle < 0 \Rightarrow f(v) < f(u_k) \quad \forall v \in K \tag{2.20}$$

setting $d_k = v - u_k$ one obtains the following condition for the stepping direction:

$$\langle \nabla f(u_k), d_k \rangle < 0 \tag{2.21}$$

this means that the direction d_k must form an acute angle with $-\nabla f(u_k)$.
Such a direction is called descent direction and condition (2.21) is necessary for all descent algorithms. A valid direction that conforms with (2.21) is shown in figure (2.3).
Note that every direction d_k that satisfies (2.21) is sufficient for the minimizing sequence $\{u_k\}$ with $u_{k+1} = u_k + td_k$ to converge to the minimum if it exists and if t satisfies (2.16). However, the convergence speed depends critically on the choice of such direction.

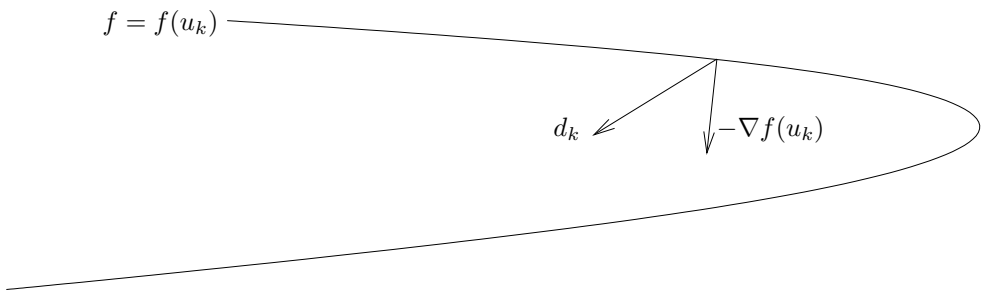


Figure 2.3: A valid descent direction forms an acute angle with the negative gradient vector.

Equation (2.21) also tells us that the directional derivative of f at u_k in the direction d_k must be negative in order for the objective f to decrease. In fact the number $\langle \nabla f(u_k), d_k \rangle$ is exactly the rate of decrease.
The challenge of defining the steepest descent direction can be stated as an optimization

problem of the following form:

$$\text{argmin}\{\langle \nabla f(u_k), d_k \rangle \quad : \quad ||d_k|| = 1\}. \tag{2.22}$$

That is, the descent direction d_k is a step of unit norm that minimizes the directional derivative and gives the largest decrease in the linear approximation of f. The norm $||\cdot||$ represents at this point a general norm and not necessarily the Euclidean norm $|\cdot|$. The normalization constraint in (2.22) is necessary since otherwise the affine function $\langle \nabla f(u_k), d_k \rangle$ would be unbounded below and the problem would be less sensible to the actual direction d_k.

2.5 First-Order Methods

As indicated above, the norm in equation (2.22) can be arbitrary. If we choose the norm *a priori* and independent of f, we obtain so-called First-Order methods. In the following we will discuss two prominent choices for the norm $||\cdot||$ in (2.22): the norm $||\cdot||_1$ and the Euclidean norm $||\cdot||_2$.

2.5.1 Choosing the (ℓ_1)-norm

The (ℓ_1)-norm is defined by:

$$||u||_1 = \sum_{i=1}^{n} |u_i| \tag{2.23}$$

The choice of the l_1- norm for problem ((2.22)) gives us the following form:

$$\text{argmin}\{\langle \nabla f(u_k), d_k \rangle \quad : \quad ||d_k||_1 = 1\}. \tag{2.24}$$

A normalized steepest descent direction in (ℓ_1) is easily characterized and can be given in a closed form. Let i be the index such that

$$||\nabla f(x)||_\infty = |\nabla f(u)|_i \tag{2.25}$$

Then the following d_k represents an optimal direction for problem (2.24):

$$d_k = -sign\,(\nabla f(u)_i)\, e_i, \tag{2.26}$$

where e_i denotes the i-th unit vector. In other words the steepest descent algorithm in the (ℓ_1)-context seeks for a component of $\nabla f(u)$ with a maximum absolute value and changes the value of the component of u_k by $-sign|\nabla f(u)_i|$.

Figure (2.4) shows that the direction d_k is restricted to stay in the diamond shaped (ℓ_1) unit ball while it goes as far as possible towards the direction $-\nabla f(u_k)$. This method is also called coordinate descent since one coordinate at a time is updated. Namely the one on which f decreases the most. The behavior of the coordinate descent method somehow reminds us of the Gauss-Seidel iteration. In fact the coordinate descent method is a variant of Gauss-Seidel for the equation $\nabla f(u_k) = 0$, where the updated variable u_k is not chosen cyclically but according to the index i defined in (2.25). The convergence of the coordinate descent method is extremely slow and gets worse with increasing dimension n. A first-order method with a convergence independent of n is presented next.

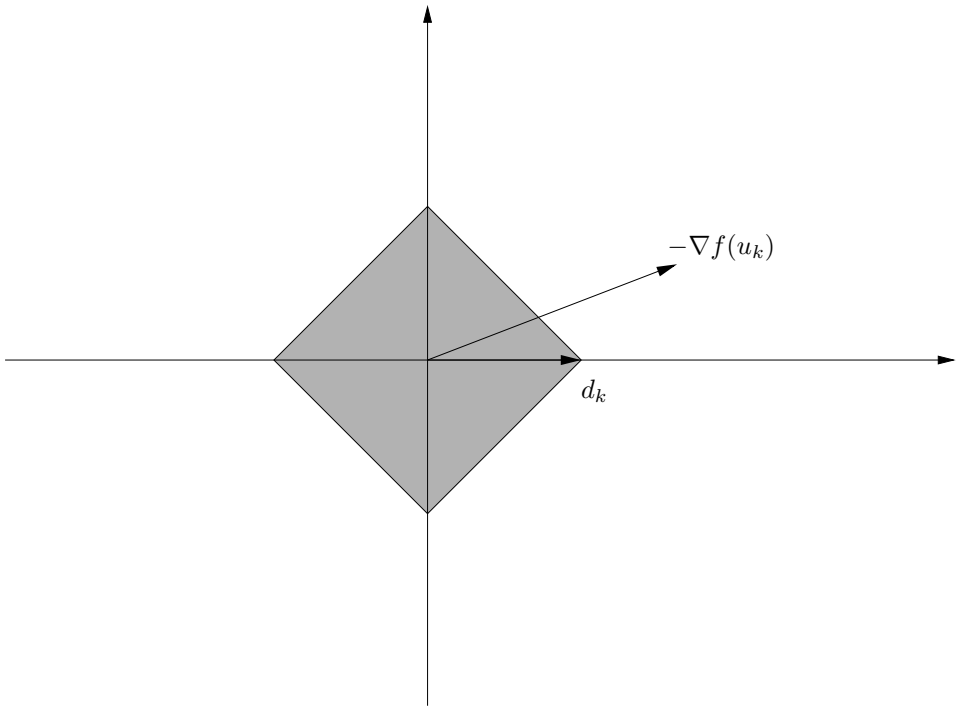


Figure 2.4: **The coordinate descent direction.**

2.5.2 Choosing the (ℓ_2)-norm

Now we consider the norm induced by the scalar product $\langle \cdot, \cdot \rangle$ namely the Euclidean norm. The Euclidean norm of a vector u is given by:

$$||u||_2 = \langle u, u \rangle^{\frac{1}{2}} \tag{2.27}$$

The method associated with the (ℓ_2)-norm is referred to as the *gradient descent method* or simply the gradient method. Thus we restate the problem of finding the direction to obtain the following form:

$$\text{argmin}\{\langle \nabla f(u_k), d_k \rangle \quad : \quad ||d_k||_2 = 1\}. \tag{2.28}$$

The geometrical interpretation of the normalized steepest descent direction d_k at u_k is that it goes as far as possible in the direction $-\nabla f(u_k)$ while staying in the unit sphere induced by the chosen norm. In the case of the norm $|| \cdot ||_2$ this means that d_k is the normed negative gradient vector:

$$d_k = -\frac{\nabla f(u_k)}{||\nabla f(u_k)||} \tag{2.29}$$

Note that normalization is obsolete because our interest lies in the actual direction and less in the step length which will be computed anyway in the line search procedure. Geometrically speaking the gradient descent direction is orthogonal to the levelsets of the objective f.

The Convergence of the Gradient Descent Method

The following definition is useful in order to describe the speed of convergence for the gradient descent method.

Definition 1 *Let u_k be a minimizing sequence that converges to u^*. The convergence of u_k is said to be* Q-linear *if there exists a constant $r \in (0,1)$ such that:*

$$\frac{||u_{k+1} - u^*||}{||u_k - u^*||} \leq r \quad \text{for all } k \text{ sufficiently large.}$$

This implies that the distance to the optimal point u^* decreases by at least the constant factor r. The Q in *Q-convergence* stands for the quotient of the successive distances to u^*.
Near the optimum the gradient method exhibits linear convergence in function values. In fact the following theorem (see [52]) gives an upper bound for the rate of decrease near the optimum u^*.

Theorem 2 *Let $f : \mathcal{R}^n :\rightarrow \mathcal{R}$ be in C^2 and let the minimizing sequence u_k generated by the gradient descent method be of optimum step length t_k in every iteration k, and u_k converges to u^*. Then:*

$$\frac{|f(u_{k+1}) - f(u^*)|}{|f(u_k) - f(u^*)|} \leq \left(\frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}}\right)^2 = r, \tag{2.30}$$

where λ_{min} and λ_{max} are the maximum and respectively the minimum eigenvalue of the Hessian matrix $\nabla^2 f(u^*)$.

The bound r is very useful in this form since it is posed in terms of the spectral condition number $\kappa(\nabla^2 f(u^*)) = \lambda_{max}/\lambda_{min}$. The condition number of the Hessian at the optimal point u^* gives a hint of how the levelsets in the area around it are shaped. With increasing κ the discrepancy between λ_{max} and λ_{min} increases leading to a smaller rate of decrease r. A high condition number of the local Hessian $\nabla^2 f(u^*)$ causes the levelsets around u^* to become more elongated, forcing the negative gradient direction to point away from the minimum. Thus the sequence of iterates is subject to zigzags leading to a slowdown of the gradient descent procedure. The effect of the condition number on the convergence of the gradient descent algorithm is illustrated in figure (2.5) for two quadratic functions with different conditioning. Note how in the case of the well conditioned function (a) the direction of negative gradient points more towards u^* than in the degenerate case (b) where $-\nabla f(u_k)$ is a bad choice for a search direction.

Although the gradient descent algorithm takes the direction of maximum decrease of the objective f, it can be very slow even for problems with moderate condition numbers say $\kappa = 1000$. For ill-conditioned problems (with high condition numbers) like the ones arising from discretizing partial differential equations e.g the discretization of the Euler-Lagrange-equation, the steepest descent method in (ℓ_2) converges unacceptably slow and should never be used.
From the computer science point of view the gradient descent method is a greedy algorithm in the sense that instead of considering future steps it looks merely at the current iteration and chooses the step of most decrease without minding the behavior of f. Next we present first-order optimization methods that exhibit much better convergence rates than the methods mentioned above.

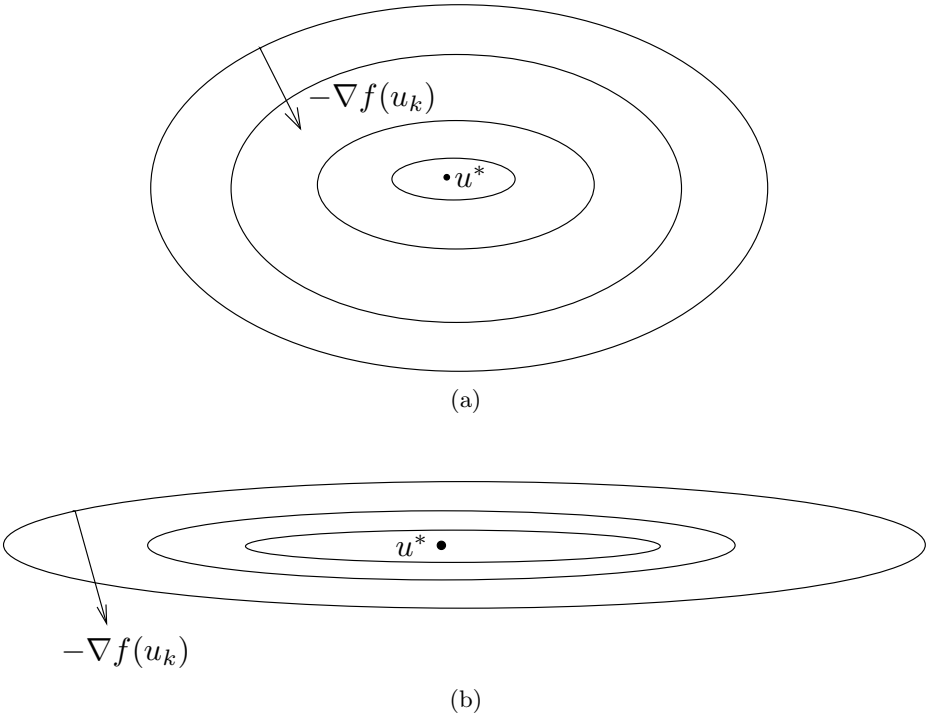


Figure 2.5: Gradient descent direction for a well conditioned (a) and a poorly scaled problem (b).

2.5.3 Improvements in First-order Methods

As discussed above, the gradient descent method uses a linear approximation of the objective and exhibits therefore slow convergence near the minimizer. In fact it has overall the following convergence behavior in the sense of complexity analysis:

$$\mathcal{F}(u_k) - \mathcal{F}(u^*) \approx \mathcal{O}\left(\frac{1}{k}\right) \tag{2.31}$$

where k denotes the iteration counter. This means that in order to achieve an accuracy of ϵ the number of iterations needed is of order $\mathcal{O}(1/\epsilon)$ which is slow and as it turns out, suboptimal for first-order methods.

The Nesterov Acceleration

Nesterov stated in [48] the following theorem:

Theorem 3 *The "optimal" first-order method for minimizing a smooth function cannot converge faster than*

$$\mathcal{F}(u_k) - \mathcal{F}(u^*) \approx \mathcal{O}\left(\frac{1}{k^2}\right). \tag{2.32}$$

in function values.

This means that an algorithm using only gradient and function evaluations cannot achieve a better rate of convergence. Thus an optimization algorithm that possesses such an optimal convergences rate needs not more than $\mathcal{O}(1/\sqrt{\epsilon})$ iterations to solve an optimization

problem to an accuracy of ϵ. Nesterov also proposed an acceleration [48,49] of the gradient descent method that achieves such an optimal rate of convergence. The only requirement on the function is being differentiable and to have a Liepschitz continuous gradient. The basic update of the Nesterov optimal method consists of the following two steps:

$$\begin{aligned} w_k &= u_{k-1} - t_k \nabla f(u_{k-1}) \\ u_k &= w_k + \frac{k-1}{k+2}(w_k - w_{k-1}) \end{aligned} \tag{2.33}$$

Hence the update is taken as a weighted sum of the previous two iterates. Note that this tactic is not heuristic and is rather theoretically motivated, for a detailed study see the following monograph ([50]) (chapter 2.2). Nesterov gives also a proof that the fixed step length

$$t_k = \frac{1}{L}$$

gives the best rate of convergence. Here the step length is taken as the reciprocal of the Liepschitz constant of the Gradient i.e.

$$||\nabla f(u) - \nabla f(v)||_2 \leq L||u - v||_2.$$

The step size depends on this Lipschitz constant since first-order methods rely on gradient information. Roughly speaking, the more Lipschitz continuos (small L) the gradient is, the better it serves as a direction, thus longer step lengths can be taken. A backtracking line search does not further improve the convergence speed. Over the years many variants of the optimal gradient descent method appeared that also solve constrained and nonsmooth optimization problems [4, 6]. In the next chapter a constrained variant of Nesterov's method, that solves the dual ROF problem, is presented.

The Conjugate Gradient Method

Another first-order method with improved theoretical convergence compared to the gradient descent method, is the *conjugate gradient* (CG) method. The CG method was originally developed for solving large linear systems of equations [33] with positive definite coefficient matrices. Consequently, it solves unconstrained quadratic problems. Later an adaptation for unconstrained nonlinear optimization problems was devised in [27]. In theory the CG method converges to the solution in n steps for n-dimensional problems. However, in practice the convergence hinges heavily on the spectrum of the coefficient matrix and particularly on the distribution of the eigenvectors. Therefore preconditioning techniques are often necessary. For a complete study of CG methods see e.g. [52].

2.6 Second-Order Methods

The methods we have seen yet are of first-order nature. This means that only information from the objective and its gradient are taken into account. We indicated above that the gradient descent method exhibits slow convergence when the sublevelsets of the objective have elongated structures. As discussed above, near these optimum the sublevelsets of f are ellipsoids induced by the Hessian. Thus it seems natural to take into account the curvature information of the sublevelsets of f, when defining d_k. This suggests involving the Hessian of f into the step selection procedure.
To do so let us consider the weighted norm defined as:

$$||u||_Q = \langle Qu, u \rangle^{\frac{1}{2}} \tag{2.34}$$

where $Q \in S_{++}$ thus is symmetric positive definite. The Quadratic form of the norm allows us to state a more general definition of problem (2.22):

$$\operatorname{argmin}\{\langle \nabla f(u_k), d_k \rangle \quad : \quad \langle Q d_k, d_k \rangle = 1\}. \tag{2.35}$$

which results in the step direction:

$$d_k = -Q^{-1} \nabla f(u_k). \tag{2.36}$$

One sees immediately that substituting the identity matrix for Q yields to the Euclidean norm and consequently to the gradient descent method.

2.6.1 The Newton Method

Choosing Q to be the local Hessian matrix carries out our initial plan to involve $\nabla^2 f$ into the direction selection. Recall that in the case of the previous norms choosing the step d_k means going as far as possible in the direction $-\nabla f(u_k)$ while staying in the unit sphere induced by the chosen norm. Involving the Hessian in the norm constraint means that the vector d_k stays in the ellipsoid

$$E_H = \{u_k + d_k : \langle H d_k, d_k \rangle \leq 1\}$$

with $H = \nabla^2 f(u_k)$. Note that $H \in S_{++}$ due to the convexity of f.

Performing the steepest descent in the norm induced by the Hessian is beneficial since the ellipsoids E_H are aligned the same way the levelsets of the objective are near the optimal point u^*. This condition is advantageous because the chosen direction points more in the direction of the minimum u^* and that way zigzagging of the minimizing sequence $\{u_k\}$ is avoided. The obtained step direction d_k reads as:

$$d_k = -\nabla^2 f(u_k)^{-1} \nabla f(u_k). \tag{2.37}$$

The computation of the inverse matrix $\nabla^2 f(u_k)^{-1}$ is numerically cumbersome and should never be done explicitly. Instead it is convenient to solve the following system of linear equations:

$$\nabla^2 f(u_k) d_k = -\nabla f(u_k) \tag{2.38}$$

in every iteration, in order to obtain the step d_k. The direction defined in (2.37) and

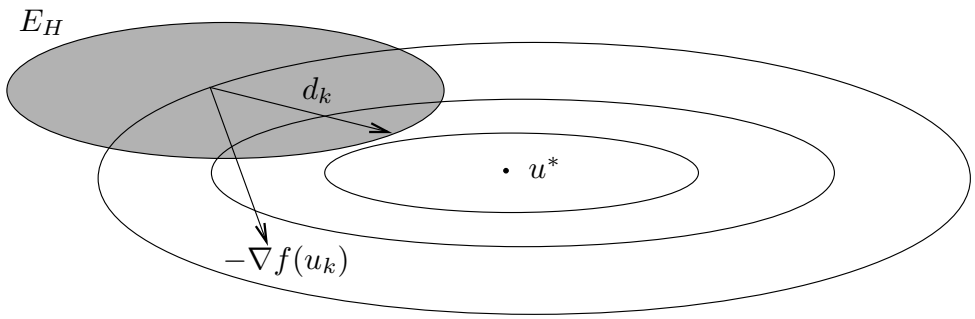


Figure 2.6: The Newton method uses the norm induced by the local Hessian to obtain a better descent direction.

equivalently in (2.38) is called the *Newton-step* of f at u_k and the corresponding algorithm is listed in algorithm 3. Figure (2.6) illustrates the Newton direction for a quadratic objective. Often Q in equation (2.35) is chosen to only approximate the true Hessian by first-order (gradient) information. In this case one speaks of *Quasi Newton Methods*.

Algorithm 3 Newton algorithm.

Given an initial point $u_o \in domf$

1. Compute the step length by solving solving for d_k where:
$$\nabla^2 f(u_k) d_k = -\nabla f(u_k)$$
2. Choose a step size t.
3. Update. $u_{k+1} = u_k + t d_k$.
4. If stopping criterion not true set $k = k + 1$ goto (1).

In what follows, two further ways to motivate the Newton method are presented:

Finding the root of the nonlinear equation $\nabla f(u^*) = 0$:

Linearizing the optimality condition $\nabla f(u^*) = 0$ near u_k leads to:

$$\nabla f(u_k + d_k) \approx \hat{f} = \nabla f(u_k) + \nabla^2 f(u_k) d_k = 0. \tag{2.39}$$

For the linearized optimality condition $f(u_k + d_k) = 0$ to hold the direction d_k has to be exactly the Newton step $d_k = -\nabla^2 f(u_k)^{-1} \nabla f(u_k)$. For a one-dimensional function $f : \mathbb{R} \to \mathbb{R}$ this leads us to the well known interpretation of the Newton method namely that of seeking the solution u^* to the problem $f'(u) = 0$ in which the update $u_k + d_k$ is the zero crossing of first-order approximation (2.39) of f' at u_k or the tangent of f' at u_k as illustrated in figure (2.7).

Minimization of the second-order approximation :

Let $\hat{f}$ be the second-order Taylor model of f at u_k:

$$\hat{f}(u_k + d_k) = f(u_k) + \langle \nabla f(u_k), d_k \rangle + \frac{1}{2} \left\langle d_k, \nabla^2 f(u_k) d_k \right\rangle \tag{2.40}$$

$\hat{f}(u_k + d_k)$ is a quadratic function of d_k.

Setting the gradient of $\hat{f}(u_k + d_k)$ to zero:

$$\begin{aligned} \nabla \hat{f}(u_k + d_k) &= \nabla f(u_k) + \nabla^2 f(u_k) d_k = 0. \\ \Leftrightarrow d_k &= -\nabla^2 f(u_k)^{-1} \nabla f(u_k). \end{aligned}$$

results again in the Newton direction for the direction d_k. This derivation also shows us that the Newton step is reliable especially if the difference between the quadratic

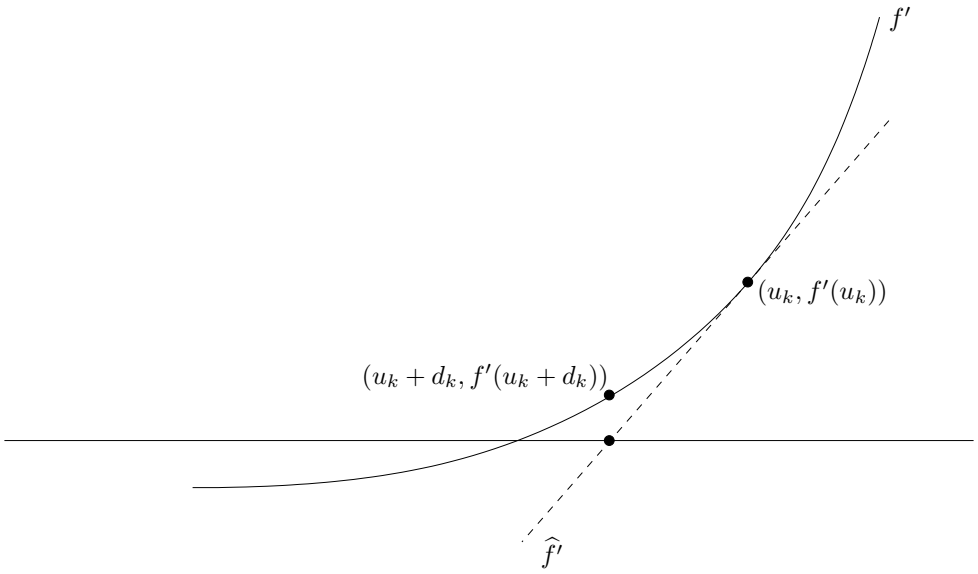


Figure 2.7: $\widehat{f}'$ is the linear approximation of f' at u_k. Adding the Newton step d_k to u_k leads to the root of $\widehat{f}'$.

model $\hat{f}(u_k + d_k)$ and the function $f(u_k + d_k)$ is not too big. This property is particularly true near the optimal point u^*. Additionally, and unlike the gradient descent method, one sees that the Newton direction comes with *a natural step length of one.*

2.6.2 The Convergence of the Newton Method

As already mentioned above the Newton step gives a good approximation if the function is similar to its local quadratic expansion. This is especially true near the minimizer of f which suggests fast *local* convergence of the Newton method.
This insight hints that the Newton direction is well suited for a function whose Hessian varies slowly or mathematically speaking is Lipschitz continuous i.e

$$||\nabla^2 f(u) - \nabla^2 f(v)||_2 \leq L||u - v||_2. \tag{2.41}$$

The Lipschitz constant L is a bound on the third derivative of f and measures how well it can be approximated by a quadratic model i.e how fast the Newton method converges for this function. In fact if $L \approx 0$ f is quadratic and the second-order Taylor approximation is exactly the function itself. In that case the Newton method minimizes f in a single step.

In order to characterize the convergence behavior of the Newton method, the notion of Q-quadratic convergence is necessary:

Definition 2 *Let a sequence* $\{u_k\}$ *converge to* u^*. *The sequence convergence is* Q-quadratic *if*

$$\frac{||u_{k+1} - u^*||}{||u_k - u^*||^2} \leq r \quad \textit{for all } k \textit{ sufficiently large}$$

with $r > 0$.

Roughly speaking, this means the number of exact digits doubles at each iteration, which is extremely fast. The Q-quadratic convergence is also known as 2^{nd} order convergence.

The classical convergence analysis of the Newton method differentiates between the following two phases:

The damped phase: Far away from the solution point u^* the quadratic model of f may not be accurate enough. In this stage most iterations require an adjusted (damped) step length $t < 1$ in case the full step of one does not cause a decrease of the objective.

The quadratically convergent phase: This is the phase where the Newton method shows its potential to minimize the objective f rapidly. In this stage f is well represented by its quadratic model and the Newton method can take full steps towards the minimizer. This explains the local 2^{nd} order convergence behavior of this method.

For a detailed discussion and rigorous proofs on the above convergence results see [52, 58].

2.6.3 Semi-Smooth Newton Methods

Optimization problems often involve functions which are nondifferentiable at certain points. Figure (2.8) shows an example of a non-smooth function which has a kink at its optimal point x^*. An important class of problems is nonlinear complementary functions. These

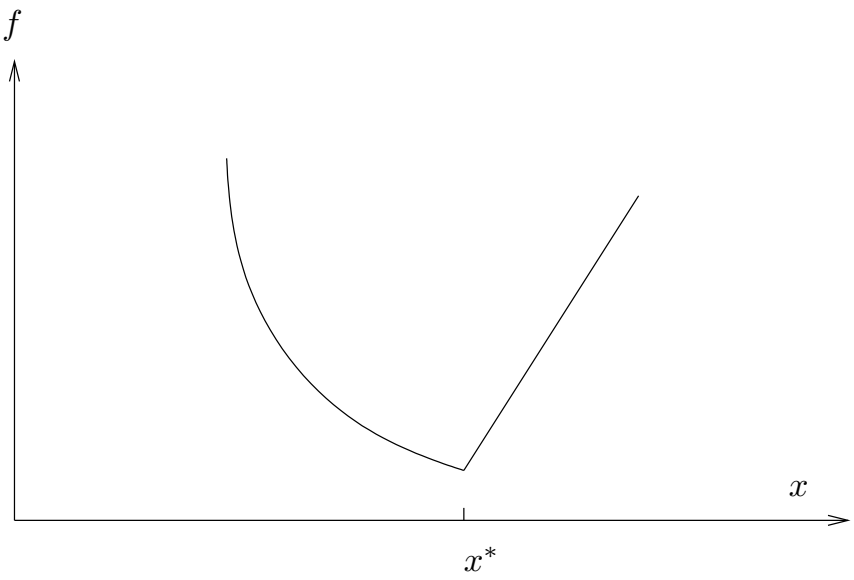


Figure 2.8: Function f has a kink at its minimum x^*.

functions express complementary conditions as nonsmooth operator equations, i.e.

$$F(x) = 0 \tag{2.42}$$

Complementary conditions emerge from variational inequalities and the complementary assumption in the Karush Kuhn Tucker conditions which we will deal with in chapter 3. The authors in [21] introduced a generalized concept of differentiability called "slant differentiability". A slight adaptation of this notion was given in [34] where the authors defined the term "Newton differentiability".

Definition 3 *Let X, Y and Z be Banach spaces and $F : D \subset X \to Z$ be a nonlinear mapping. The mapping F is called "Newton differentiable" on $U \subset D$ if there exists a linear mapping $G : U \to \mathcal{L}(X, Z)$ such that:*

$$\lim_{h \to 0} \frac{1}{||h||} ||F(x+h) - F(x) - G(x+h)h|| = 0 \tag{2.43}$$

for every $x \in U$.

If F is a Newton differentiable mapping, equation (2.42) can be tackled by Newton's method. The mapping G serves as a generalized derivative that can be involved in the Newton method. Take for instance the point-wise maximum and minimum operators in a real valued function space X (say L^2) on the domain $\Omega \subset \mathbb{R}^n$, which we will use several times in this work. In [34] it is shown that the operators $max(0, \varphi)$ and $min(0, \varphi)$ with $\varphi \in X$ are Newton differentiable and the following mappings are candidates for a generalized derivative G that satisfies equation (2.43) for all $\varphi \in X$:

$$G_{max}(\varphi)(x) = \begin{cases} 1 & \text{if } \varphi(x) \geq 0, \\ 0 & \text{if } \varphi(x) < 0; \end{cases} \tag{2.44}$$

and

$$G_{min}(\varphi)(x) = \begin{cases} 1 & \text{if } \varphi(x) < 0; \\ 0 & \text{if } \varphi(x) \geq 0. \end{cases} \tag{2.45}$$

In order to describe the local convergence properties of semi-smooth Newton methods, the following definition is necessary :

Definition 4 *Let a sequence $\{u_k\}$ converges to u^*. The sequence convergence is Q-superlinear if*

$$\lim_{k \to \infty} \frac{||u_{k+1} - u^*||}{||u_k - u^*||} = 0.$$

Superlinear convergence is placed between linear and quadratic convergence. Note that every quadratic convergent sequence is also superlinear convergent and that every superlinear convergent sequence complies with linear convergence.

The generalized Newton method converges superlinearly when $||u_0 - u^*||$ is sufficiently small. For proofs on convergences results we advise the reader to see [21]. Semismooth Newton methods are in contrary to their smooth counterparts not descend methods, hence the function values are not necessarily decreased by every iterate.

Chapter 3

Newton Methods for TV Minimization

3.1 Introduction

The ongoing research into TV minimization has its reasons in the non-differentiability of the TV semi-norm as well as in its highly nonlinear associated Euler-Lagrange equation. Another difficulty is the large scale nature of the underlying optimization problem since it has to be solved for every pixel. These challenges triggered intense research activities from which numerous algorithms arisen. The aim of this chapter is to review some of the optimization methods for solving the ROF model. However, the focus will lie on second-order methods and particularly Newton's method. We begin this chapter by introducing notations and discretization techniques. We then review the dual formulation of the ROF model. As mentioned the algorithmic part of this chapter treats mainly Newton methods. Nonetheless a lot of progress has been made in other classes of optimization algorithms to solve the ROF problem. A review of these techniques is therefore due and will include:

- First-order methods.
- Graph-cut techniques.

Afterwards, second-order methods are presented and will be subdivided into the following categories:

- The SOCP approach.
- Newton methods applied to the primal ROF formulation.
- Newton methods that solve the primal-dual ROF formulation.
- Newton methods that solve the dual ROF formulation.

We conclude this chapter by proposing a Newton method which solves the dual formulation a Huber regularized ROF model. The special structure of the Huber regularizer helps reducing an artifact known from pure TV regularization, namely the staircasing effect. Moreover we show that the Huber regularization is equivalent with a Tikhonov regularization of the dual problem of the original ROF model. In order du solve the constrained dual problem, we devise an efficient method that solves the underlying KKT-system by means of an active set strategy.

3.2 Discretization and Notations

In order to implement variational methods on a digital computer by means of convex optimization techniques, one has to discretize the underlying partial differential equations which are derived in the continuous setting. The discretization of the Euler-Lagrange equation is twofold: finding a suitable finite vector space to discretize the image domain and discrete operator representations for the differential operators ∇ and div which preserves important features known from their continuous counterparts.

3.2.1 Discretization of the Image Domain

For simplicity we assume the image domain Ω, to be in the unit square $[0,1] \times [0,1]$ and discretize it by a $n \times n$ grid. We then address each pixel by its corresponding index pair (i,j) which represents the points $(i/(n+1), (j/(n+1)) \in \Omega$. Thus the discrete image will be assumed to live in the vector space $X = \mathbb{R}^{n \times n}$. Furthermore, we denote the discrete vector field $\Omega \to \mathbb{R}^2$ by $Y = X \times X$ and endow X and Y with the Euclidean scalar products:

$$\begin{aligned} \langle u, v \rangle_X &= \sum_{1 \le i,j \le n} u_{i,j} v_{i,j} \\ \langle p, q \rangle_Y &= \sum_{1 \le i,j \le n} \left(p^1_{i,j} q^1_{i,j} + p^2_{i,j} q^2_{i,j} \right) \end{aligned}$$

for $u, v \in X$ and $p, q \in Y$.

3.2.2 Discretization of the Differential Operators

Optimization problems feature typically an operator that lies in the objective and that relates the variables sought in the process. These operators are in their canonical form matrices. In case of variational methods these matrices emerge from discretizing differential operators. The discretized differential operator should restore the circumstances in the continuous setting such as boundary conditions and the relation to the adjoint operator, in order to produce good approximate results. In the following the point-wise discretization of the differential operators is discussed before getting to the representation of the derivatives in a matrix-vector multiplication framework.

Discretization of the Gradient

For every pixel (i,j) the vector valued gradient operator $\nabla u : X \to Y$ with forward differences is given as:

$$(\nabla u)^1_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < n \\ 0 & i = n. \end{cases} \tag{3.1}$$

$$(\nabla u)^2_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < n \\ 0 & j = n. \end{cases} \tag{3.2}$$

Note that for $i, j = n$ the Neumann boundary conditions $\frac{\partial u}{\partial n} = 0$ hold.

Discretization of the Divergence Operator

In order to clarify the relation between the operators ∇ and div, the distributional definition of the gradient comes into play. In equation (1.5) we see that $-\text{div}$ is the adjoint operator of ∇ i.e.

$$\langle \nabla u, p \rangle_Y = \langle u, -\text{div } p \rangle_X \Longleftrightarrow \nabla^* = -\text{div}. \tag{3.3}$$

This insight gives us the following formulation for the discrete divergence operator of a vector field p:

$$(\text{div } p)_{i,j} = \begin{cases} p^1_{i+1,j} - p_{i,j} & \text{if } 1 < i < n \\ p^1_{i,j} & \text{if } i = 1 \\ -p^1_{i-1,j} & \text{if } i = n \end{cases} + \begin{cases} p^2_{i,j} - p_{i,j-1} & \text{if } 1 < j < n \\ p^2_{i,j} & \text{if } j = 1 \\ -p^2_{i,j-1} & \text{if } j = n \end{cases}.$$

Hence the div operator is defined by backward differences using Dirichlet boundary conditions. These boundary conditions emphasize the compact support property of the test functions p.

3.2.3 Representation as a Matrix-vector Product

The discrete formulae we obtained so far for the differential operators do not suggest their usage in a convex optimization framework. Thus it seems natural to express the linear operators ∇ and div as matrix operators in order to describe a variational problem in matrix algebra language. Since matrices operate on vectors we stack the image u row-wise into a vector $\hat{u}$. The indices of the matrix $u \in \mathbb{R}^{n \times n}$ and of the vector $\hat{u} \in \mathbb{R}^N$ with $N = n^2$ relate as follows:

$$\hat{u}_{(j-1)n+i} = u_{i,j}, \;\; 1 \leq i, j \leq n. \tag{3.4}$$

Let us call the map (3.4) Π_N which works as follows:

$$\Pi_N(u) = \hat{u}.$$

As for vector fields we proceed similar to the scalar case. Let $p = (p_1, p_2) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n}$. We stack first p_1 then p_2 row-wise in a new vector $\hat{p} \in \mathbb{R}^{2N}$ hence:

$$\Pi_{2N}(p1, p2) = \hat{p} = \begin{pmatrix} \Pi_N(p_1) \\ \Pi_N(p_2) \end{pmatrix}.$$

Let us for convenience denote the Euclidean norm on the entries of $\hat{p}$ with $|\hat{p}|_i = \sqrt{\hat{p}_i + \hat{p}_{i+N}}$ for all $1 \leq i \leq N$. In what follows we describe how to construct the ∇ operator for two dimensions with help of the one dimensional forward difference operator (3.1) which we denote by ∇_1. The matrix representation of the operator $\nabla_1 \in \mathbb{R}^{n \times n}$ can be sketched as follows:

$$\nabla_1 = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ & & & & 0 \end{bmatrix}. \tag{3.5}$$

Next we give an example of the effect of ∇_1 on a vector $v \in \mathbb{R}^4$ that represents a one-dimensional signal:

$$\nabla_1 = \begin{bmatrix} -1 & 1 & & \\ & -1 & 1 & \\ & & -1 & 1 \\ & & & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} v_2 - v_1 \\ v_3 - v_2 \\ v_4 - v_3 \\ 0 \end{bmatrix}. \tag{3.6}$$

From the resulting vector one sees that forward differences are realized and that the boundary conditions occuring in (3.1) are preserved. The generalization of ∇_1 to two dimensions for a $n \times n$ image can be realized as follows:

$$\nabla_2^n = \begin{bmatrix} I_n \otimes \nabla_1 \\ \nabla_1 \otimes I_n \end{bmatrix} \in \mathbb{R}^{2N \times N} \tag{3.7}$$

where $\otimes$ denotes the Kronecker product and I_n the n-dimensional identity matrix. If we write out the matrix (3.7) we see that ∇_2 looks as follows:

$$\nabla_2 = \begin{bmatrix} \nabla_1 & & & & & \\ & \nabla_1 & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & & \nabla_1 \\ -I_n & I_n & & & & \\ & -I_n & I_n & & & \\ & & \ddots & \ddots & & \\ & & & -I_n & I_n & \\ & & & & & O_n \end{bmatrix}$$

where O_n stands for the $n \times n$ zero matrix. It is easy to see that a multiplication with an image vector $\hat{u}$, as defined in (3.4), gives us for the upper half of ∇_2 the differences in the x-direction and for the lower half respectively the differences in the y direction. Hence:

$$\begin{bmatrix} I_n \otimes \nabla_1 \\ \nabla_1 \otimes I_n \end{bmatrix} \hat{u} = \begin{pmatrix} \nabla_x \hat{u} \\ \nabla_y \hat{u} \end{pmatrix}.$$

Since we now know the identity $\nabla^* = -\text{div}$ the representation of the divergence operator is straightforward:

$$\text{div} = -\begin{bmatrix} I_n \otimes \nabla_1 \\ \nabla_1 \otimes I_n \end{bmatrix}^T = -\begin{bmatrix} I_n \otimes \nabla_1 & \nabla_1 \otimes I_n \end{bmatrix} \in \mathbb{R}^{N \times 2N} \tag{3.8}$$

since the adjoint of a real valued matrix is simply its transpose. Note that in order to change the type of discrete operator (say to central differences), we simply modify matrix ∇_1 to the desired difference scheme and perform the same procedure as in (3.7). To simplify notations we denote our derivative operators in the following manner:

$$A^T = \nabla$$

and

$$-A = \text{div}$$

with $A = \begin{bmatrix} I_n \otimes \nabla_1 & \nabla_1 \otimes I_n \end{bmatrix}$. This matrix-vector notation is very useful for formulating and implementing the Newton systems which we encounter in this chapter. However, often the point-wise description of the systems is more comprehensive. Therefore, unless otherwise stated we will make use of the point-wise notation when formulating problems.

3.3 Duality

The first ones to study the concept of duality for the Total Variation image model were Chan, Golub and Mulet in [19]. The dual formulation of the ROF model has drawn much attention in the imaging community lately. This is because of its nice structure that is better suited for convex optimization techniques. The discrete version of the ROF model reads as:

$$\min_{u\in X} ||\nabla u||_Y + \frac{1}{2\lambda}||u-f||_X^2. \tag{3.9}$$

With $||p||_Y = \langle p,p\rangle_Y^{1/2}$ and $||p||_X = \langle p,p\rangle_X^{1/2}$. In what follows the dual formulation of the ROF model is derived without loss of generality in a discrete setting. As a starting point we rewrite the TV term as:

$$||\nabla u||_Y = \sup_{p\in\mathcal{V}} \langle p, \nabla u\rangle_Y = \sup_{p\in\mathcal{V}} \langle -\text{div } p, u\rangle_X \tag{3.10}$$

with

$$\mathcal{V} = \{p \in Y : |p_{i_j}| \leq 1, \ \ \forall i,j\}$$

which is an immediate consequence of the Cauchy-Schwarz inequality. Substituting equation (3.10) in the primal ROF model (3.9) one obtains the following problem:

$$\min_u \max_{p\in\mathcal{V}} \Psi(u,p) := \langle p, \nabla u\rangle_Y + \frac{1}{2\lambda}||u-f||_X^2 \tag{3.11}$$

This is the so-called primal-dual problem and is essentially a saddle point problem. Observe that $\Psi(u,p)$ is strict convex in u and concave (linear) in p. This justifies using the max-min theorem (see [56]) which allows us to interchange the max and min in functional (3.11):

$$\max_{p\in\mathcal{V}} \min_u \Psi(u,p) := \langle p, \nabla u\rangle_Y + \frac{1}{2\lambda}||u-f||_X^2. \tag{3.12}$$

The inner minimization problem can then be eliminated by differentiating $\Psi(u,p)$ with respect to u, setting $\nabla_u \Psi(u,p) = 0$ and substituting the optimal value for u in equation (3.11). The optimal function for u reads as:

$$\nabla_u \Psi(u,p) = 0 \Leftrightarrow u^* = f + \lambda \text{div} p.$$

The saddle-point problem consequently becomes

$$\begin{aligned} &\min_{p\in\mathcal{V}} \langle \text{div } p, f\rangle_X + \frac{\lambda}{2}||\text{div } p||_X^2 \\ \Leftrightarrow &\min_{p\in\mathcal{V}} \ \frac{1}{2\lambda}\left(||\lambda \text{div } p + f||_X^2 - ||f||_X^2\right) \end{aligned}$$

which is equivalent to the so called dual formulation of the ROF model:

$$\min_{p\in\mathcal{V}} \frac{1}{2}||\lambda \text{div} p + f||_X^2. \tag{3.13}$$

For convenience we stated the above equation as a minimization problem by changing the sign of the functional. The dual problem is quadratic and thus less non-linear than the unconstrained ROF model. Additionally, and opposed to the primal problem, it is continuously differentiable and does not suffer from the singularity $|\nabla u| = 0$. The only drawback are the many nonlinear constraints that occur in every point $i,j \in \Omega$. This fact poses optimization-wise a challenge.
Once the optimal dual variable p^* is obtained the solution u^* to the ROF problem reads as:

$$u^* = f + \lambda \text{div} p^*. \tag{3.14}$$

3.3.1 The KKT-conditions

Dealing with an inequality constrained problem the first-order optimality conditions derive from the Karush-Kuhn-Tucker conditions [38]. To state the KKT (Karush-Kuhn-Tucker) conditions the Lagrangian of (3.13) is necessary and is given by:

$$\mathcal{L}(p,\alpha) = \frac{1}{2}||\lambda \text{div} p + f||^2 + \sum_{i,j} \alpha_{i,j}|p_{i,j}|^2. \tag{3.15}$$

Here $\alpha \in X$ denotes the Lagrange multiplier where $\alpha_{i,j}$ is associated with the constraint $|p_{i,j}| \leq 1$. Let p^* and α^* be the primal and dual[1] optimal points of the Lagrangian. The KKT conditions state that for $\mathcal{L}(p^*, \alpha^*)$ the following must hold for all $i, j = 1, ..., n$:

- The primal optimal variable $p^*_{i,j}$ has to comply with the constraint hence $|p^*_{i,j}|^2 \leq 1$. This property is also called feasibility.
- The optimal $\alpha^*_{i,j}$ Lagrange multiplier exists and $\alpha^*_{i,j} \geq 0$ must hold.
- The gradient of $\mathcal{L}(p^*, \alpha^*)$ with respect to p must vanish at p^*:

$$\left(\frac{\partial \mathcal{L}(p^*, \alpha^*)}{\partial p}\right)_{i,j} = -(\nabla(\lambda \text{div}\, p^* + f))_{i,j} + \alpha^*_{i,j} p^*_{i,j} = 0.$$

- The product of the Lagrange multiplier and the constraint is zero:

$$\alpha^*_{i,j}(|p^*_{i,j}|^2 - 1) = 0.$$

  Additionally, p^* and α^* have to fulfill the so-called *complementary slackness* condition which gives us the following case differentiation:

  Case 1: $\alpha^*_{i,j} \geq 0 \Longrightarrow |p^*_{i,j}| = 1$
  Case 2: $\alpha^*_{i,j} = 0 \Longrightarrow |p^*_{i,j}| < 1$

  In the first case the constraint is said to be active and in the latter it is called inactive.

Chambolle makes in his work the following key observation. For both cases, active and inactive constraint ,the Lagrange multiplier can be written as:

$$\alpha_{i,j} = |(\nabla(\lambda \text{div}\, p + f))_{i,j}|\,. \tag{3.16}$$

Now the Lagrange multiplier can be eliminated from the optimality equation and one obtains :

$$-(\nabla(\lambda \text{div}\, p + f))_{i,j} + |(\nabla(\lambda \text{div} p + f))_{i,j}|\, p_{i,j} = 0. \tag{3.17}$$

From this equation Chambolle derives a fixed point method for solving the dual formulation of ROF. This so-called semi-implicit gradient descent method will be discussed in the next section.

[1] The reader should be cautious to not confuse the duality in this context of constrained programs with the notion of duality applied to the total variation.

3.4 First-Order Methods

The major characteristic of first-order methods is that they only need functional and gradient evaluations. Therefore, they require little memory and are additionally simple to implement. This makes them well-suited for large-scale problems and implementation on parallel computing architecture, such as graphics processing units, possible. The main drawback of first-order methods is the slow rate of convergence. However, this slow convergence speed is compensated by the very low complexity of each iteration. In the following we review some gradient based methods for TV minimization. We divide these methods into primal, dual and primal-dual methods.

3.4.1 Time Marching Scheme

In section (1.2.1) a time marching scheme to solve the perturbed ROF model was presented. Equation (1.17) is proposed by ROF in their original paper [57] and is equivalent to the steepest descent method with a constant step length:

$$u^{k+1} = u^k - \triangle t \left[-\text{div}\left(\frac{\nabla u^k}{|\nabla u^k|_\beta} \right) + \frac{1}{\lambda}(u^k - f) \right] \tag{3.18}$$

of the energy function. Note that the time step $\triangle t$ is exactly the step length in the optimization context. An upper bound for the time step is the so-called *Courant-Friedrichs-Lewy* (CFL) condition

$$\triangle t \leq c \triangle x^2 |\nabla u| \tag{3.19}$$

with some constant c. The space step $\triangle x$ is usually assumed to be 1. This condition is necessary for convergence while solving PDE's numerically. This poses a serious restriction on the step length especially in flat regions where $|\nabla u| \approx 0$, leading to very slow convergence.
In [44] Marquina and Osher use a preconditioning technique to relax the CFL condition and to cancel out the singularity $\frac{1}{|\nabla u|}$. They obtain the following step:

$$\frac{\partial u}{\partial t} = |\nabla u| \left[-\text{div}\left(\frac{\nabla u}{|\nabla u|_\beta} \right) + \frac{1}{\lambda}(u - f) \right] \tag{3.20}$$

by multiplying $|\nabla u|$ to the original one. The step length is now independent of the magnitude of the gradient. However, this is a severe modification of the original ROF model which makes the preconditioning technique questionable.

3.4.2 Duality Based Methods

The dual formulation of ROF on the contrary to the original problem needs not to be smoothed with an artificial parameter. This allows its minimization to converge to the true optimizer. Although we get rid of the non-singularity of the primal formulation of ROF we pay the tribute by dealing with many constraints that occur in every pixel. The naive application of the gradient descent method, by taking the negative gradient as the direction, would fail since there is no mechanism preventing the iterate to exit the feasible set. Note that line search methods will eventually produce a step length even, if the variable violates the constraint, and thus does not guarantee feasibility. Many classical techniques for solving constrained non-linear programs exist such as interior point methods, barrier methods and augmented Lagrangian methods. In [10] Carter applied an interior point method and a barrier relaxation method for solving the dual ROF model. Unfortunately, these algorithms suffer from heavy runtime.

The Projected Gradient Method

Luckily, although the constraints are numerous, they are spatially separable and can be conquered by a simple point-wise projection on the feasible set. Chambolle [13] was the first to take advantage of this fact by introducing a re-projection force to the gradient descent method. The so-called *projected gradient* method is given by the following update:

$$p_{k+1} = P_C\left(p_k + \frac{\tau}{\lambda}[\nabla(f + \lambda \text{div} p_k)]\right) \tag{3.21}$$

The inner update is a simple gradient descent with step length $\frac{\tau}{\lambda}$. The outer operator $P_C(p)$ is a projection on the set $C = \{p : |p_{i,j}| \leq 1\}$ which can be cheaply computed as follows:

$$(P_C(p))_{i,j} = \frac{p_{i,j}}{max\{1, |p_{i,j}|\}}. \tag{3.22}$$

The gradient projection method converges empirically for step lengths with $\tau \approx \frac{1}{4}$ and is a much faster method than the time marching scheme (3.18).

Chambolle presented in [12] a slightly faster method. From equation (3.17) he derives a fixed point method which consists of the following updates:

$$(p_{k+1})_{i,j} = \frac{(p_k)_{i,j} + \frac{\tau}{\lambda}(\nabla(\lambda \text{div}\, p_k + f))_{i,j}}{1 + \frac{\tau}{\lambda}|(\nabla(\lambda \text{div}\, p_k + f)_{i,j}|} \tag{3.23}$$

This method is similar to the projected gradient formulation given in equation (3.21). In the same work it was shown that the fixpoint equation (3.23) is a contraction[1] only if $\tau \leq 1/8$. However, the sequence converges in practice for step-sizes up to $1/4\lambda$.

The gradient projection algorithm can be interpreted as an instance of *operator splitting* for minimizing a sum of two convex functions.

$$\min_u \{\mathcal{F}(p) = f(p) + g(p)\} \tag{3.24}$$

If we take $f(p) = \frac{1}{2}|\lambda \text{div} p + f|^2$ to be the dual objective and $g(p) = I_C$ to be the indicator function of the unit disc, we recover exactly the dual ROF problem. The Euclidean projection $P_C(p)$ then corresponds to the so-called proximal operator of $g(p)$ which is essentially an implicit gradient descent step of $g(p)$. For details see e.g [2, 6, 11] and references therein. In [5] Beck and Teboulle also showed that the above gradient projection algorithm belongs to a class of algorithms that exhibit a sublinear rate of convergence in functional values i.e.

$$\mathcal{F}(p_k) - \mathcal{F}(p^*) \approx \mathcal{O}\left(\frac{1}{k}\right), \tag{3.25}$$

where k is the iteration counter. Hence, it suffers from the same slow convergence of the gradient descent method which, as we highlighted in section (2.5.3), is suboptimal.

The method of Beck and Teboulle

In section (2.5.3) an improved gradient descent method devised by Nesterov in [48] is presented. This is done by exploiting not only the information of the current gradient but also the gradient information of previous iterations.

[1] By the Banach fixed point theorem every contraction in a complete metric space admits a unique fixpoint.

Inspired by Nesterov's algorithm, Beck and Teboulle devised recently in [5] a variant for nonsmooth problems and gave a proof for an optimal convergence rate of $\mathcal{O}(1/k^2)$. The so-called *Fast Iterative Shrinkage Algorithm* (FISTA) uses in each iteration not only local gradient information but also exploits gradient information of the two previous iterates. The FISTA paradigm applied on the projected gradient algorithm consists of the steps listed in algorithm (4). P_C denotes the same projection procedure as in equation (3.22).

Algorithm 4 The FISTA algorithm.

- Initialize $u_0 := f$, $\psi_0 = 0$, $p_0 = 0$ $t_0 = 1$ and $k = 0$
- Iterate:

$$\begin{aligned} u^{k+1} &= f + \lambda \text{div } p^k \\ \psi^{k+1} &= P_C\left(p^k + \frac{1}{8\lambda}\nabla u^{k+1}\right) \\ tk+1 &= \frac{1+\sqrt{1+4t_k^2}}{2} \\ p^{k+1} &= \psi^{k+1} + \frac{t_k - 1}{t^{k+1}}(\psi^{k+1} - \psi^k) \end{aligned}$$

The new p_{k+1} iterate is updated in a sophisticated linear combination of the previous two iterates.
The step size $1/8\lambda$ here, as well as in the previous dual algorithms, is taken as the reciprocal of the Lipschitz constant L of the gradient $-\nabla(f + \lambda \text{div} p)$. The Lipschitz constant is an upper bound on the derivative, hence it can be written by means of the operator norm of the Hessian $-\lambda\nabla\text{div}$ of the dual objective:

$$L = \lambda||\nabla\text{div}|| = \lambda||\text{div}^*\text{div}|| = \lambda||\text{div}||^2 \leq 8\lambda \tag{3.26}$$

thus $L \leq 8\lambda$. The right inequality is proven by Chambolle in [12] for the discretized divergence operator. Today the method of Beck and Teboulle is the most efficient first-order method for TV minimization.

3.4.3 Primal-Dual Methods

We have seen that the difficulty in solving the primal problem via gradient descent lies in the non-smoothness of regions $x \in \Omega$ where $|\nabla u(x)| = 0$. Whereas the dual problem suffers particularly in regions where the constraint is active hence, $|p(x)| = 1$ and consequently $|\nabla u(x)| \gg 0$. From this one deduces that both formulations suffer from complementary difficulties. This fact encouraged researchers to look into methods that combine both the primal and the dual formulation in order to cancel out the difficulties of one another. Primal-Dual methods tackle the saddle point problem (3.11) and seek for a saddle point (p^*, u^*) by a gradient descent in u and a gradient ascent in p. Thus one has in every iteration the following gradient ascent/descent steps:

$$\begin{aligned} p_{k+1} &= p_k + \theta\nabla_p\Psi(u_k, p_k) \\ u_{k+1} &= u_k - \tau\nabla_u\Psi(u_k, p_{k+1}) \end{aligned}$$

where Ψ denotes the functional from equation (3.11) and θ, τ stand for some step sizes. In the literature these alternating ascent/descent methods are also referred to as Arrow-Hurwicz algorithms [3]. Such approach was proposed by Zhu and Chan in [66] and is listed in algorithm (5). The authors apply a re-projection P_C as in equation (3.22) in order to force the dual variable to stay feasible. The step sizes τ_k and θ_k are, in contrast to the

Algorithm 5 The Primal-Dual Algorithm by Zhu and Chan.

- Initialize $p_0 = 0$ $k = 0$ and $u_0 = f$
- Iterate

$$\begin{aligned} \tau_k &= 0.2 + 0.08k \\ \theta_k &= \left(0.5 - \frac{5}{15+k}\right)\tau_k \\ p_{k+1} &= P_C\left(p_k + \theta_k \nabla u_k\right) \\ u_{k+1} &= u_k + \tau_k\left(\text{div } p_{k+1} + \frac{1}{\lambda}(f - u_k)\right) \end{aligned}$$

previous algorithms, being varied in every iteration. However, no proof of convergence for this method, with heuristic adjustment of the step sizes, is given. Pock, Bischof, Cremers, and Chambolle proposed, in their recent work [54] on minimizing the piecewise constant Mumford-Shah functional, a variant of the primal-dual algorithm. Their algorithm is inspired by a paper of Popov [55] and is suited for a larger class of problems. The authors also give a proof of convergence for fixed step sizes.

3.5 Graph-Cut Based Methods

Recently also graph-cut based approaches for minimizing the ROF model have been studied (see e.g. [13,62]). These approaches are based on the observation that an image can be decomposed in its super-level sets. The idea is a direct consequence of the co-area formula presented in section (1.1.5) which states that the Total Variation sums up the perimeters of all upperlevel-sets, hence:

$$\int_\Omega |Du| = \int_{-\infty}^{\infty} Per(E_\gamma) d\gamma \tag{3.27}$$

and that minimizing TV results in minimizing the perimeter of the sets

$$E_\lambda = \{(x) \in \gamma |\ u(x) > \gamma\}$$

for all γ in the range of u. In the discrete case these supper-level sets are finite. In fact one has for an L bit grey valued image 2^L possible grey values e.g. $L = 8$ results in 256 grey values. The discrete co-area formula then reads as:

$$J_{TV}(u) = \sum_{\gamma=0}^{2^L-2} J_{TV}\left(\chi\left\{u \geq \gamma\right\}\right) \tag{3.28}$$

J_{TV} denotes some discretization of the Total Variation energy and $\chi\left\{u \geq \gamma\right\}$ stands for the characteristic function of the super-level set $\{u \geq \gamma\}$ thus:

$$\chi\{u \geq \gamma\}(x) = \begin{cases} 1 & \text{if } u(x) \geq \gamma \\ 0 & \text{else} \end{cases}$$

Now suppose J_{TV} is a discretization of TV that fulfills the discrete co-area formula (3.28) and let u^* be the unique solution to the ROF model:

$$\min_{u \in X} J_{TV}(u) + \frac{1}{2\lambda}||u - f||^2 \tag{3.29}$$

then for all $\gamma \geq 0$ the characteristic functions $\chi\{u^* \geq \gamma\}$ and $\chi\{u^* > \gamma\}$ are respectively the largest and smallest minimizer of:

$$\min_{\theta \in \{0,1\}} \lambda J_{TV}(\theta) + \sum_{i,j} \theta_{i,j}(\gamma - f_{i,j}) \tag{3.30}$$

If $\chi\{u \geq \gamma\} = \chi\{u > \gamma\}$ then the solution of (3.30) is unique. The characteristic functions $\chi\{u \geq \gamma\}$ and $\chi\{u > \gamma\}$ are different if $\gamma = u_{i,j}$ for any $(i,j) \in \Omega$. This connection between the discrete ROF model and problem (3.30) is stated and proven by Chambolle in [13] and by Darbon and Sigelle in [62], and is a discrete instance of theorem (1). Problem (3.30) is a binary shape optimization problem where

- $J_{TV}(\theta)$ measures the perimeter of the shape given implicitly by θ and involves pairwise interactions.
- The second term represents a spatially separable fidelity measure thus without any interconnections involved.

Being a binary labeling Markov random field, problem (3.30) can be solved efficiently using *graph-cut* algorithms. As a consequence the discretized ROF model (3.29) can be solved exactly by minimizing the binary shape optimization problem (3.30) for every $\gamma \in 0, ..., L-2$.
Since the duality theorem of Ford and Fulkerson, it is well known that finding the minimum cut of a graph is equivalent to finding the maximal flow along its edges. For a detailed study we refer to [8]. The maximal flow can be computed in polynomial time which suggests that the ROF model would be solved rapidly. In practice, maximal flow can be solved in linear [9] (dependent on the number of pixels) runtime. As emphasized above, the discretized ROF (3.29) model can be minimized by means of graph-cut algorithms only if $J_{TV}(u)$ satisfies the discrete co-area formula (3.28). Unfortunately, this is not true for the isotropic discretization of TV:

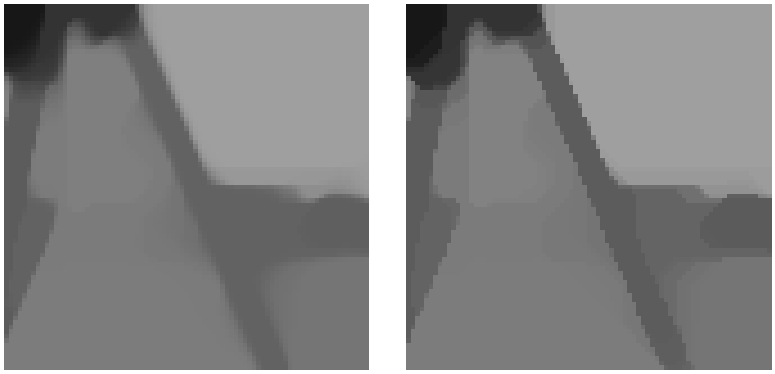
$$J_{iso}(u) = \sqrt{(u_{i+1,j} - u_{i,j})^2 + (u_{i,j+1} - u_{i,j})^2} \tag{3.31}$$

Functions satisfying the discrete co-area formula also fulfill a property called *sub-modularity* [13]. In [40] it was shown that only sub-modular pairwise interactions functions of binary variables can be minimized via graph-cut algorithms. An example for a discretization of J_{TV}, that satisfies equation (3.28) and is consequently sub-modular, is:

$$J_{aniso}(u) = |u_{i+1,j} - u_{i,j}| + |u_{i,j+1} - u_{i,j}| \tag{3.32}$$

This discretization is a measure of perimeter in *Manhattan-metric* and would result in a 4-connected graph. More complex interaction functions are also possible, provided they are sub-modular. The advantage of the Graph-Cut approach is that it yields an exact solution to the ROF problem up to a machine precision. The major disadvantage of graph-cut algorithms is that due to so-called metrication errors, edges in particular orientations are favored. Although, more complex discretizations can be chosen, i.e with a higher connectivity to approximate the isotropic TV, the resulting reconstruction remains

(a) Noisy image (b) Continuous ROF (c) Graph Cut ROF



(d) Continuous ROF (e) Graph cut ROF

Figure 3.1: A comparison between a continuous reconstruction (b) and a graph cut reconstruction (c) of the noisy cameraman image (a). A magnification shows the crystalline structure due to metrication artifacts (e) of the graph-cut based ROF reconstruction compared to the smoother continuous pendant (d).

somehow cristalline. Figure (3.1) illustrates graph-cut image denoising and compares the result with a reconstruction via a continuous algorithm. The graph-cut algorithm[2] uses a 16-connected graph and produces close result to the continuous output. However such high connectivity leads to a heavy memory requirement, in particular when a 3D application is considered.

From the convex optimization point of view, solving the ROF model, by means of graph-cut algorithms, corresponds to a sequence of linear programs. Whereas the continuous methods solve constrained or unconstrained non-linear optimization problems.

[2]The implementation is taken from the website of Chambolle: http://www.cmap.polytechnique.fr/ antonin/software/

3.6 Second-Order Cone Programming

Goldfarb and Yin present in [31] an algorithm for solving the constrained formulation of the ROF model which can be reformulated in the following discretized fashion:

$$\begin{aligned} J_{TV}(u) &= \sum_{1\leq i,j\leq n} |(\nabla u)_{i,j}| \\ s.t. \qquad & ||u-f||^2 \leq \sigma^2 \end{aligned} \tag{3.33}$$

Their approach formulates the ROF model as a so-called *second-order cone program* (SOCP). A SOCP is an optimization problem with second-order cone constraints. An n-dimensional second-order (quadratic) cone is the following set of vectors:

$$\mathcal{K}^d = \left\{(x,t) \in \mathbb{R}^{d-1} \times \mathbb{R} \quad s.t. \ |x|_2 \leq t\right\} \tag{3.34}$$

and is the norm cone associated with the ℓ_2 norm. A d-dimensional second-order cone constraint is of the form:

$$(x,t) \in \mathcal{K}^d$$

In order to reformulate the constrained ROF model to a SOCP, the authors introduce new variables p, q, t and v. The additive noise is represented by the variable v hence $v = f - u$. The varibles p and q represent the components of the discrete forward difference operator ∇:

$$p_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < n \\ 0 & i = n. \end{cases} \tag{3.35}$$

$$q_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < n \\ 0 & j = n. \end{cases} \tag{3.36}$$

If one demands that for every pixel the variables $(p_{i,j}, q_{i,j}, t_{i,j})$ lie in a 3-dimensional second-order cone i.e.

$$\left[(p_{i,j}, q_{i,j})^T, t_{i,j}\right] \in \mathcal{K}^3 \Longleftrightarrow \sqrt{p_{i,j}^2 + q_{i,j}^2} \leq t \tag{3.37}$$

then definition (3.34) tells us that t is an upper bound on $|(p_{i,j}, q_{i,j})^T|$ and respectively on $|(\nabla u)_{i,j}|$. Figure (3.2) shows the boundary of the second-order cone which contains feasible points $(p_{i,j}, q_{i,j}, t)$. This constraint suggests that minimizing the sum of $t_{i,j}$ amounts to minimizing the discrete TV in equation (3.33). The noise constraint in equation (3.33) can also be formulated as the following second-order cone constraint:

$$(\sigma, v) \in \mathcal{K}^{n^2+1}$$

The following constrained optimization problem sums up the above facts:

$$\begin{array}{lll} \min & \sum_{i,j} t_{i,j} & \\ s.t & u_{i,j} - v_{i,j} = f_{i,j} & \text{for } i,j = 1,...,n \\ & -p_{i,j} + u_{i+1,j} - u_{i,j} = 0 & \text{for } i = 1,...,n-1,\ j = 1,...,n \\ & -q_{i,j} + u_{i+1,j} - u_{i,j} = 0 & \text{for } i = 1,...,n,\ j = 1,...,n-1 \\ & p_{n,j} = 0 & \text{for } j = 1,...,n \\ & p_{i,n} = 0 & \text{for } i = 1,...,n \\ & \left[(p_{i,j}, q_{i,j})^T, t_{i,j}\right] \in K^3 & \text{for } i,j = 1,...,n \\ & (\sigma, v) \in \mathcal{K}^{n^2+1} & \end{array}$$

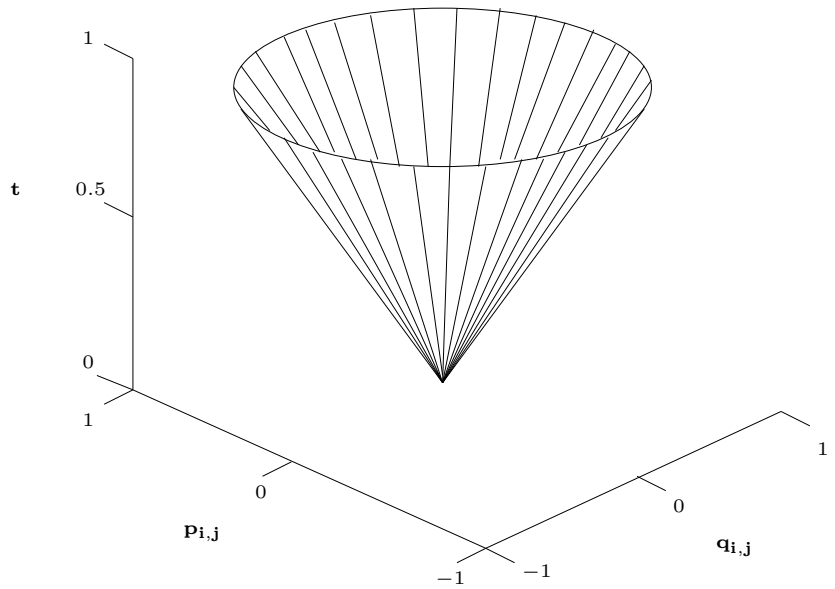


Figure 3.2: The boundary of the second-order cone $\mathcal{K}^3$. The second-order cone constraint is fulfilled only if the vector $(p_{i,j}, p_{i,j}, t)$ lies in the ice-cream cone.

Note the similarity of problem (3.38) with linear programs as both have linear objectives. In fact it turns out that, like in the case of linear programs, SOCP's can be solved by interior point methods in polynomial time. The main reference on interior-point methods for SOCP's is the book by Nesterov and Nemirovsky [65]. Interior point methods solve essentially a Newton linearization of a primal-dual system in every iteration and have therefore similar convergence properties to the Newton methods discussed in the next sections. In order to solve the SOCP, Goldfarb and Yin make use of a commercial SOCP solver called MOSEK[3]. The authors also cast the dual ROF formulation as a SOCP.

3.7 Primal Newton Methods

The convenient way to solve the ROF model via Newton's method is to linearize the Euler-Lagrange equation. This is done by tackling the primal formulation of the ROF model:

$$F(u) = \int_\Omega |\nabla u| \; dx + \frac{1}{2\lambda} \int_\Omega (u - f)^2 dx. \tag{3.38}$$

Its Euler-Lagrange equation is the following non-linear partial differential equation:

$$G(u) = -\text{div}\left(\frac{\nabla u}{|\nabla u|}\right) + \frac{1}{\lambda}(u - f) = 0 \tag{3.39}$$

with Neumann boundary condition $\frac{\partial u}{\partial n} = 0$ on $\partial\Omega$. The singularity at $\nabla(u) = 0$ can be dealt with by adding a small parameter $\beta > 0$ to the norm of the gradient. One ends up minimizing the following energy functional:

$$F_\beta(u) = \int_\Omega |\nabla u|_\beta \; dx + \frac{1}{2\lambda} \int_\Omega (u - f)^2 dx.$$

[3] http://www.mosek.com/

where $|\nabla u|_\beta = \sqrt{|\nabla u|^2 + \beta}$, by solving the perturbed Euler-Lagrange equation:

$$G_\beta(u) = -\text{div}\left(\frac{\nabla u}{|\nabla u|_\beta}\right) + \frac{1}{\lambda}(u - f) = 0 \tag{3.40}$$

The formal Newton step to solve (3.40) is then given by:

$$J(u_k)d_k = -G_\beta(u_k) \tag{3.41}$$

Where $J(u_k)$ denotes the Jacobi matrix of G_β and respectively the Hessian of F_β at u_k. $J(u)$ is given by:

$$J(u) = -\text{div}\left(\frac{1}{|\nabla u|_\beta}\left(I - \frac{\nabla u \nabla u^T}{|\nabla u|_\beta^2}\right)\nabla\right) + \frac{1}{\lambda} \tag{3.42}$$

As discussed in section (1.2.1) the constant β should not be too large, in order for the unconstrained ROF model to keep the edge information in its admitted solution. For the Newton method to converge we have seen in the convex optimization chapter (section (2.6.2)) that the Hessian must be Lipschitz continuos. The Lipschitz constant L of the Hessian $J(u_k)$ depends on the value of β and is expected to increase when F_β approaches non-differentiability as $\beta \to 0$. In fact in [63] Vogel and Oman observed that L behaves like β^{-3} when β is small (i.e $\ll 1$). This implies that the pure Newton method would suffer deterioration if β is chosen small. Vogel and Oman showed in their numerical results that Newton's method exhibits a small domain of convergence for when β is small, hence it needs many iterations to reach its quadratic convergence phase. In the same work the authors also compared the formal Newton method to the gradient descent method and a proposed relaxed fixed point method in which we get into detail next.

3.7.1 The Fixed Point Method of Vogel and Oman

Vogel and Oman propose in [63] a fixed point method for solving equation (3.40). The new iterate $u_{(k+1)}$ is obtained by solving the following system of equations for u_{k+1}:

$$-\text{div}\left(\frac{\nabla u_{k+1}}{|\nabla u_k|_\beta}\right) + \frac{1}{\lambda}(u_{k+1} - f) = 0 \tag{3.43}$$

in each iteration until convergence occurs. Note that the nonlinearity of the Euler-Lagrange equation that lies in the diffusivity coefficient $|\nabla u_{k+1}|_\beta$ is removed by considering the diffusivity $|\nabla u_k|_\beta$ evaluated at the previous iterate. This is the reason why the method is also called *lagged diffusivity fixed point method*. On first sight this method would not be classified as a second-order method. However, Vogel and Oman showed that by setting $d_k = u_{k+1} - u_k$ we end up with the following Quasi-Newton step:

$$A(u_k)d_k = -F_\beta \tag{3.44}$$

where $A(u_k)$ approximates the true Jacobian J(u) and is given by:

$$A(u) = -\text{div}\left(\frac{\nabla}{|\nabla u|_\beta}\right) + \frac{1}{\lambda}. \tag{3.45}$$

Comparing $J(u)$ with $A(u)$ one observes that one gets from the former to the latter by dropping the higher order term

$$\frac{\nabla u \nabla u^T}{|\nabla u|_\beta}.$$

In the numerical experiments performed by the authors the method (3.43) proved to not require any globalization techniques to control the step length. Furthermore, it is less prone to the choice of β compared to the pure Newton method. However, the convergence speed is still linear theoretically, although in practice it proved to produce visually good results after a few iterations.

3.7.2 The Continuation Method of Chan et al.

As indicated in the last section, Newton's method performs poorly due to the highly nonlinear term $\nabla u/|\nabla u|_\beta$. The difficulty is that the local quadratic approximation which is the basis for the Newton method is not a good model for the higher nonlinear primal formulation of Total Variation. Near the minimizer Newton's method converges very fast so it seems natural to try an initial guess near to the minimum. This is achieved by Chan et al. [20] by combining the Newton method with a continuation procedure. This is done by using a solution for problem (3.40) with a large perturbation parameter β as the initial guess, to solve the same problem with a smaller β and gradually reduce β to the desired value.

3.8 Primal-Dual Newton Methods

Primal-dual problems in Total Variation minimization are posed as an objective of the primal variable u and the dual variable p. Essentially primal-dual Newton algorithms solve the optimality condition of saddle-point problem (3.11), given as the following non-linear system of equations:

$$-\text{div } p + \frac{1}{\lambda}(u - f) = 0 \tag{3.46}$$

$$p|\nabla u| - \nabla u = 0 \tag{3.47}$$

or slight modifications of it. Thus these algorithms seek for saddle-points of the max / min problem (3.11). They relate to the first-order primal-dual methods as both algorithm classes solve the primal-dual objective (3.46). Solving equation (3.46) by means of Newton's method is less problematic since the (p,u) system is less non-linear in u than the primal ROF formulation. We have seen in the last chapter that the Newton method in its pure form finds the root of the equation $\nabla f(u) = 0$. Hence it does not distinguish between minima, maxima or saddle points and is therefore suitable in this case.

3.8.1 The Algorithm of Chan Golub and Mulet

As pointed out, the Newton method applied on the primal formulation of the ROF model does not work satisfactorily due to the presence of a highly nonlinear term in the optimality system which causes a small convergence domain. To overcome this problem Chan, Golub and Mulet (CGM) use a Primal-Dual method in order to linearize the Euler-Lagrange equation (3.40). The idea of the CGM method is to remove the singularity caused by the non-differentiability of the Euclidean norm by introducing a *dual flux* variable p to the optimality system with:

$$p = \frac{\nabla u}{|\nabla u|_\beta} \tag{3.48}$$

This yields to the following equivalent system of nonlinear partial differential equations:

$$\begin{aligned} F(p,u) &= -\text{div } p + \frac{1}{\lambda}(u-f) = 0 \\ G(p,u) &= p|\nabla u|_\beta - \nabla u = 0 \end{aligned} \tag{3.49}$$

Note that the second equation is exactly the identity (3.48). The above system is also referred to as the primal-dual system and is a perturbed version of equation (3.46). The Jacobi matrix of the (u,p) system (3.49) is obtained by the following differentiation:

$$\begin{bmatrix} (\nabla F(p,u))^T \\ (\nabla G(p,u))^T \end{bmatrix} = \begin{bmatrix} \frac{\partial F(p,u)}{\partial p} & \frac{\partial F(p,u)}{\partial u} \\ \frac{\partial G(p,u)}{\partial p} & \frac{\partial G(p,u)}{\partial u} \end{bmatrix} \tag{3.50}$$

which yields to the following linearization of equation system (3.49):

$$\begin{bmatrix} -\text{div} & \frac{1}{\lambda} \\ H & -\left(I - \frac{p(\nabla u)^T}{|\nabla u|_\beta}\right)\nabla \end{bmatrix} \begin{bmatrix} \delta p \\ \delta u \end{bmatrix} = \begin{bmatrix} -F(p,u) \\ -G(p,u) \end{bmatrix}. \tag{3.51}$$

Matrix I denotes the two by two identity matrix. Matrix H denotes the 2×2 diagonal matrix with the entries $H_{11} = H_{22} = |\nabla u|_\beta$. The Jacobi matrix in this case is three times bigger than the linearization of the primal system which results in a much higher computational cost. Luckily, the system of equations has a particular block structure and makes the usage of the Schur complement method (see appendix A) possible. The Schur complement of H is the following equation:

$$S = \left[-\text{div}\left(\frac{1}{|\nabla u|_\beta}\left(I - \frac{p\nabla u^T}{|\nabla u|_\beta}\right)\nabla\right) + \frac{1}{\lambda}\right]. \tag{3.52}$$

This way CGM were able to eliminate the δp from the system to obtain a much smaller one which only depends on δu. Since H is a diagonal coefficient matrix its inverse is simply H^{-1} with $H_{11}^{-1} = H_{22}^{-1} = 1/|\nabla u|_\beta$. The solution vector is then obtained by solving for δu the following system of equations:

$$S\delta u = \text{div}\left(\frac{\nabla u}{|\nabla u|_\beta}\right) - u + f. \tag{3.53}$$

The dual step $\delta\hat{p}$ can then be obtained by backsubstituting δu in the lower equation in (3.51) leading to a closed formula for δp:

$$\delta p = \frac{1}{|\nabla u|_\beta}\left(I - \frac{p(\nabla u)^T}{|\nabla u|_\beta}\right)\nabla\delta u - p + \frac{\nabla u}{|\nabla u|_\beta} \tag{3.54}$$

Note that matrix:

$$C = \frac{p(\nabla u)^T}{|\nabla u|_\beta} = \frac{1}{|\nabla u|_\beta}\begin{pmatrix} p_1(\nabla u)1 & p_1(\nabla u)^2 \\ p_2(\nabla u)^1 & p_2(\nabla u)^2 \end{pmatrix} \tag{3.55}$$

is not symmetric ($p_1(\nabla u)^2 \neq p_2(\nabla u)^1$) therefore the conjugate gradient (CG) method cannot be applied to the system (3.53). CGM propose the following symmetrization technique:

$$\widetilde{C} = \frac{1}{2}\left(C + C^T\right) = \frac{1}{2}\frac{p(\nabla u)^T + (\nabla u)p^T}{|\nabla u|_\beta}. \tag{3.56}$$

This leads to the modified equation

$$\left[-\text{div}\left(\frac{1}{|\nabla u|_\beta}\left(I-\widetilde{C}\right)\nabla\right)+\frac{1}{\lambda}\right]\delta u=\text{div}\left(\frac{\nabla}{|\nabla u|_\beta}\right)-u+f \tag{3.57}$$

for obtaining the step δu and respectively

$$\delta p=\frac{1}{|\nabla u|_\beta}\left(I-\widetilde{C}\right)\nabla\delta u-p+\frac{\nabla u}{|\nabla u|_\beta} \tag{3.58}$$

as a formula for δp. The authors justify the symmetrization since matrix C converges to the symmetric modification (3.56). This is based on the fact that p converges to $\frac{\nabla u}{|\nabla u|}$ towards the minimum, hence in the limit p satatisfies:

$$p(x)=\frac{\nabla u(x)}{|\nabla u(x)|}$$

for allmost all $x\in\Omega$ with $\nabla u(x)\neq\vec{0}$. CGM also point out that the underlying matrix of the system of equations (3.53) is only positive definite if the dual variable p is restricted to the unit sphere i.e. $|p|\leq 1$. This constraint can be assured by a simple line search where the step length t is chosen such that:

$$t=\delta\sup\{\alpha:|p_{i,j}+\alpha\delta p_{i,j}|<1\ \ \forall i,j\}\qquad\delta\in(0,1) \tag{3.59}$$

The complete algorithm is listed in algorithm 6. Note that the CGM algorithm does not

Algorithm 6 The CGM algorithm.

1. Initialize (u_0,p_0) and set k=0
2. Solve equation (3.57) for δu and compute δp via equation (3.58)
3. Compute a step length t for δp_k that satisfies (3.59)
4. Update variables

$$\begin{aligned}u_{k+1}&=u_k+\delta u\\p_{k+1}&=p_k+t\delta p\end{aligned}$$

5. Stop, or set $k:=k+1$ and go to step 2

require a line search on the primal variable u and is expected to converge very fast in practice. Furthermore, it was shown that the convergence rate deteriorates only moderately with decreasing β.

In [67] Zhu investigates the connection between solving the SOCP formulation of the constrained ROF model introduced in section (3.6) and tackling the primal-dual system by means of Newton's method. The KKT-system of the SOCP problem is solved via interior points methods. Zhu relates the smoothing constant β with the barrier parameter used for solving the SOCP KKT-system. The barrier parameter in interior point methods is reduced in every step. Borrowing the same idea the author proposes a method that dynamically reduces β in every step of the CGM method. The resulting algorithm handles very small values for β and makes a convergence arbitrary close to the true minimizer possible.

3.8.2 The Algorithm of Hintermüller and Stadler

The algorithms we have seen so far, use a smoothing constant that is not selective to whether or not smoothing is necessary. Thus it seems natural to perform local smoothing in areas where the gradient is too small and causes numerical difficulties, and to leave out points where the gradient has a large magnitude and is more likely to agree with an edge. This approach was applied in the work of Hintermüller and Stadler (HS) in [35]. HS derive a similar approach to the CGM method where smoothing is only applied to certain points in Ω. HS derive the following semismooth system of optimality equations:

$$\begin{aligned} \widetilde{F}(p,u) &= -\text{div } p + \frac{1}{\lambda}(u-f) = 0 \\ \widetilde{G}(p,u) &= \max(\gamma, |\nabla u|)p - \nabla u = 0 \end{aligned} \tag{3.60}$$

The semismoothness of this equation lies in the semismooth natur of the $max(\cdot,\cdot)$ operator . The linearization of equation (3.60) poses the challenge of differentiating the non-differentiable max operator. To address this problem the authors make use of the generalized derivative defined in equation (2.44):

$$G_{max}(\varphi)(x) = \begin{cases} 1 & \text{if } \varphi(x) \geq 0, \\ 0 & \text{if } \varphi(x) < 0; \end{cases} \tag{3.61}$$

Substituting for $\varphi(x) = |\nabla u(x)| - \gamma$ yields to the following generalized derivative for $\max(\gamma, |\nabla u|)$:

$$G_{max} = \begin{cases} 1 & \text{if } |\nabla u| \geq \gamma, \\ 0 & \text{else;} \end{cases} \tag{3.62}$$

The semismoothnes of the max operator justifies the application of the Newton method on equation (3.60). The resulting equation for the Newton step reads as:

$$\begin{bmatrix} -\text{div} & \frac{1}{\lambda} \\ \widetilde{H} & -G_{max}\left(I - \frac{p(\nabla u)^T}{|\nabla u|}\right)\nabla \end{bmatrix} \begin{bmatrix} \delta p \\ \delta u \end{bmatrix} = \begin{bmatrix} -\widetilde{F}(p,u) \\ -\widetilde{G}(p,u) \end{bmatrix} \tag{3.63}$$

With $\widetilde{H} \in \mathbb{R}^{2\times 2}$ such that $\widetilde{H}_{11} = \widetilde{H}_{22} = \max(\gamma, |\nabla u|)$. Having the same block structure as the CGM Newton System, equation (3.63) is amenable to the Schur complement alteration. The reduced system again depends only on δu:

$$\begin{aligned} \left[-\text{div}\left(\frac{1}{\max(\gamma,|\nabla u|)} G_{max}\left(I - \frac{p\nabla u^T}{|\nabla u|}\right)\nabla\right) + \frac{1}{\lambda}\right]\delta u = \\ \text{div}\left(\frac{\nabla u}{\max(\gamma,|\nabla u|)}\right) - u + f. \end{aligned} \tag{3.64}$$

The δp step is then given by:

$$\delta p = \frac{1}{\max(\gamma,|\nabla u|)} G_{max}\left(I - \frac{p(\nabla u)^T}{|\nabla u|}\right)\nabla\delta u - p + \frac{\nabla u}{\max(\gamma,|\nabla u|)}. \tag{3.65}$$

Like the CGM system the matrix $\frac{p(\nabla u)^T}{|\nabla u|}$ is not symmetric, causing incompatibility with the CG method and variants of it. This is solved similar to the above approach and with the same symmetrization scheme. One obtains the following modified Newton steps:

$$\begin{aligned} \left[-\text{div}\left(\frac{1}{\max(\gamma,|\nabla u|)} G_{max}\left(I - \widetilde{C}\right)\nabla\right) + \frac{1}{\lambda}\right]\delta u = \\ \text{div}\left(\frac{\nabla u}{\max(\gamma,|\nabla u|)}\right) - u + f. \end{aligned} \tag{3.66}$$

The δp step is then given by:

$$\delta p = \frac{1}{\max(\gamma, |\nabla u|)} G_{max} \left(I - \widetilde{C} \right) \nabla \delta u - p + \frac{\nabla u}{\max(\gamma, |\nabla u|)}. \tag{3.67}$$

In their work HS perform an eigenvalue analysis of the Newton system coefficient matrix (3.66) and come to the conclusion that positive definiteness (positive eigenvalues) of the matrix is only ensured if the variable p stays in the feasible set given by the unit circle. As mentioned above CGM deal with the problem by performing a linesearch on the step δp to make sure $p + \delta p$ stays feasible. This is a major drawback since the step length is applied to the update $p_{k+1} = p_k + t\delta p$ for every point $(i, j) \in \Omega$. Thus the step is also restricted for feasible variables. This is dealt with differently by HS, as they propose a projection of p into the unit disc before assembling the Jacobi matrix in (3.66) . This way the Jacobi matrix stays positive definite. The HS method is listed in algorithm (7). Without a restricting linesearch the dual variable p is allowed to become infeasible and

Algorithm 7 The HS algorithm.

1. Initialize (u_0, p_0) and set k=0
2. Project p into the unit disc.
3. Solve equation (3.66) for δu and compute δp via equation (3.67)
4. Update variables

$$\begin{aligned} u_{k+1} &= u_k + \delta u \\ p_{k+1} &= p_k + t\delta p \end{aligned}$$

5. Stop, or set $k := k + 1$ and go to step 2

full Newton steps are achieved. Therefore, one expects higher convergence rate than the CGM method.

3.9 Dual Newton Methods

As mentioned above the advantage of solving the dual problem is that no smoothing parameter is needed. The minimizer of the dual objective is thus the true solution of the ROF model. Another advantage is that the objective function is quadratic and is thus more amenable to Newton's method than the original ROF formulation. In this work, two strategies for solving the KKT-system of the constrained dual problem, are presented.

- The first approach involves the *Fischer-Burmeister* function. This function is tailored to non-linear complementary problems. It makes sure that the complementary conditions that occur in the KKT optimality conditions hold at its minimum.
- The second strategy is the *active set approach*. This method computes in every step an estimate of the indices for those variables for which one believes that the constraint is active hence $|p_{i,j}| = 1$. Based o n this prediction a Newton linearization of the Lagrangian formulation is solved.

Using the latter strategy, we will present in this chapter an algorithm that solves a regularized version of the dual formulation. Priorly an application of the first approach done by Ng et al. is presented in the next section.

3.9.1 The Semismooth Newton Method by Ng et al.

The Fischer-Burmeister strategy, as well as the active set method, expresses complementary conditions as nonsmooth operator equations. In order to be able to solve these equations by means of Newton's method, they need to fulfill an important property, namely semismoothness. Semismooth functions are not differentiable everywhere. To describe the variation at certain non-differentiable points, one has to calculate so-called generalized derivatives that approximate the directional derivative sufficiently. In what follows, we define semismoothness as well as the generalized derivative of semismooth functions.

Semismooth and Strongly Semismooth Functions

Semismooth functions appeared in [46] for the first time where there usability for constrained optimization was already considered. This relaxed smoothness assumption places semismooth functions between Lipschitz functions and continuously differentiable functions. In the following, an alternative formulation for the generalized derivative already mentioned in section (2.6.3) is reviewed. If a function G is locally Lipschitz then by Rademacher's theorem G is differentiable almost everywhere (except in a countable set of points, hence with Lebesgue zero measure). This property is required to make the following definitions.
Let D_G denote the points at which G is differentiable. The B-subdifferential of G at x is defined as follows:

$$\partial_b G(x) = \left\{ J : J = \lim_{x_i \to x, x_i \in D_G} \nabla G(x_i) \right\} \tag{3.68}$$

The Clarke generalized derivative [22] is defined as:

$$\partial G(x) = co\ \partial_B G(x) \tag{3.69}$$

where *co* stands for the convex hull.

The function G is called *semismooth* at x if it is locally Lipschitz at x and:

$$\lim_{V \in \partial G(x+th'), h' \to h, t \downarrow 0} Vh' \tag{3.70}$$

exists for all h. Semismooth functions are directionally differentiable. The directional derivative at x in direction h is approximated by the limit (3.70). There is even a measure of the semismoothness which describes how the directional derivative (3.70) differs from the Gâteau derivative at x in direction h. Suppose G is semismooth at x, then for all $V \in \partial G(x+h)$ and any $h \to 0$ the following holds:

$$Vh - G'(x;h) = o(|h|). \tag{3.71}$$

Even better is the following property; if G is locally Lipschitz and for any $V \in \partial G(x+h)$ and for all $h \to 0$:

$$Vh - G'(x;h) = O(|h|^2) \tag{3.72}$$

the function G is called *strongly semismooth*. Semismoothness and strong semismoothness qualify the element V defined in (3.70) to be involved in a Newton-type method since it describes sufficiently the local variation of the function and is therefore a good substitute for the Fréchet derivative.

The Fischer-Burmeister Function

Ng et al. in [51] use a nonsmooth formulation of the complementary conditions given in section (3.3.1) based on the simple function:

$$\phi(a,b) := \sqrt{a^2+b^2} - a - b. \tag{3.73}$$

ϕ is known as the Fischer-Burmeister function and was introduced by Fischer in [25]. The Fischer-Burmeister function is strongly semismooth and has the property that

$$\phi(a,b) = 0 \Leftrightarrow a \geq 0, b \geq 0, ab = 0. \tag{3.74}$$

This property paves the way to use the Fischer-Burmeister function for making sure the non-linear complementary conditions for the constraint $|p| \leq 1$ hold at the solution vector (p^*, α^*).

The KKT-conditions as Semismooth Nonlinear Equations

For the sake of convenience, we reformulate problem (3.15) to conform with the matrix-vector representation we have defined in section (3.2.3). For this purpose, we map the scalar fields $u, g, \alpha \in X$ and the vector field $p \in Y$ to $\hat{u}, \hat{\alpha}, \hat{g} \in \mathbb{R}^N$ and $\hat{p} \in \mathbb{R}^{2N}$ by means of the maps we defined in section (3.2.3) i,e $\Pi_N(u) = \hat{u}$ and $\Pi_N,(g) = \hat{g}$ and $\Pi_{2N}(p) = \hat{p}$. The optimality condition (3.16) for the Lagrangian (3.15), cast in a matrix-vector framework, reads as:

$$\frac{\partial \mathcal{L}(\hat{p}, \hat{\alpha})}{\partial \hat{p}} = (A^T(\lambda A\hat{p} + \hat{f})) + \hat{\alpha} \odot \hat{p} = 0$$

The expression $\hat{\alpha} \odot \hat{p}$ is understood as following component-wise multiplication i.e

$$\hat{\alpha} \odot \hat{p} = (\hat{\alpha}_1\hat{p}_1, \cdots, \hat{\alpha}_N\hat{p}_N, \hat{\alpha}_1\hat{p}_{N+1}, \cdots, \hat{\alpha}_N\hat{p}_{2N})^T$$

Matrix A is from section (3.2.3) i.e.

$$\begin{aligned} A^T &= \nabla \\ -A &= \text{div}. \end{aligned}$$

Now let z be a vector which is composed of the vector $\hat{p}$ and $\hat{\alpha}$, concatenated as follows:

$$z = \begin{pmatrix} \hat{p} \\ \hat{\alpha} \end{pmatrix}.$$

The KKT-system of problem (3.15) can the be stated in the following nonsmooth equations:

$$\Phi(z) = \begin{pmatrix} \frac{\partial L(\hat{p},\hat{\alpha})}{\partial \hat{p}} \\ \phi_1(1 - |\hat{p}|_1, \alpha_1) \\ \vdots \\ \phi_N(1 - |\hat{p}|_N, \alpha_N) \end{pmatrix} = 0 \tag{3.75}$$

Hence solving $\Phi(z) = 0$ amounts to solving problem (3.13).

The Modified Gauss-Newton Algorithm

Instead of solving the nonlinear system of equations (3.75) the authors tackling solving the following equation:

$$\Psi(z) := \frac{1}{2}\Phi(z)^T\Phi(z) = \frac{1}{2}||\Phi(z)||^2 = 0. \tag{3.76}$$

Ψ is the so-called natural merit function of $\Phi(z)$. This means that now one minimizes the norm of the natural residual, which is a better measure for the distance to the optimal point and hence provides a good stopping criterion. Any root z^* of Ψ is obviously also a root of ϕ.
An important property of the Fischer-Burmeister function ϕ is that ϕ^2 is continuously differentiable which guarantees the smoothness of the merit function Ψ and therefore makes it amenable to Newton-type methods for solving nonlinear least-squares problems e.g. the *Gauss-Newton* algorithm.
The special form of Ψ makes the least-squares problem easier to solve than general unconstrained minimization problems. This is based on the fact that the derivatives of Ψ can be expressed in terms of the Jacobian of Φ. The Jacobi matrix of Φ is its transposed gradient V^T. In our nonsmooth case V is an element of the generalized gradient of Φ defined in equation (3.69), hence:

$$V \in \partial\Phi(z) \tag{3.77}$$

with $\partial\Phi(z)$ as defined in (3.70). It is known (see for instance [52]) that the generelized gradient and the Hessian of Ψ can be given as follows:

$$\partial\Psi(z) = \frac{1}{2}||\Phi(z)||^2 = \partial\Phi(z)^T\Phi(z) \ni V^T\Phi(z). \tag{3.78}$$

$$H(z) \approx \partial\Phi(z)\partial\Phi(z)^T \ni VV^T \tag{3.79}$$

Here, $H(z)$ denotes the generalized Hessian of $\Psi(z)$. Note that in (3.78) the chain rule for generalized gradients is used. V^TV is a first-order approximation for the Hessian and is especially accurate near the minimum. For a detailed proof see for instance [52](Chapter 10). At this point we have all the ingredients to formulate the Newton method for problem (3.76). But first we have to calculate the Jacobian V^T of Φ. The next result gives a description of an element matrix in the generalized Jacobian $\partial\Phi(z)^T$. Each element $V^T \in \partial\Phi(z)^T$ can be written as follows:

$$V^T = \begin{pmatrix} \frac{\partial^2\mathcal{L}(\hat{p},\hat{\alpha})}{\partial\hat{p}^2} & \frac{\partial^2\mathcal{L}(\hat{p},\hat{\alpha})}{\partial\hat{\alpha}\hat{p}} \\ \left[D\left(\frac{\partial\phi_i}{\partial\hat{p}_i}\right) D\left(\frac{\partial\phi_i}{\partial\hat{p}_{i+N}}\right)\right] & D\left(\frac{\partial\phi_i}{\partial\hat{\alpha}_i}\right) \end{pmatrix} \tag{3.80}$$

where $\frac{\partial^2\mathcal{L}(\hat{p},\hat{\alpha})}{\partial\hat{p}^2} = \lambda A^TA + D(\hat{\alpha},\hat{\alpha})$ with $D(\hat{\alpha},\hat{\alpha}) \in \mathbb{R}^{N\times N}$ the diagonal matrix:

$$D(\hat{\alpha},\hat{\alpha}) = diag(\hat{\alpha}_1,...,\hat{\alpha}_N,\hat{\alpha}_1,...,\hat{\alpha}_N)$$

The derivative $\frac{\partial^2\mathcal{L}(\hat{p},\hat{\alpha})}{\partial\hat{\alpha}\hat{p}}$ has the following form:

$$\frac{\partial^2\mathcal{L}(\hat{p},\hat{\alpha})}{\partial\hat{\alpha}\hat{p}} = \begin{bmatrix} D(\hat{p}_i) \\ D(\hat{p}_{i+N}) \end{bmatrix}$$

where the diagonal matrices $D(\hat{p}_i)$ and $D(\hat{p}_{i+N})$ are given by:

$$D(\hat{p}_i) = diag(\hat{p}_1,...,\hat{p}_N)$$

$$D(\hat{p}_{i+N}) = diag(\hat{p}_{N+1}, ..., \hat{p}_{2N})$$

$D\left(\frac{\partial\phi_i}{\partial\hat{\alpha}_i}\right) \in \mathbb{R}^{N\times N}$, $D\left(\frac{\partial\phi_i}{\partial\hat{p}_i}\right)$ and $D\left(\frac{\partial\phi_i}{\partial\hat{p}_{i+N}}\right) \in \mathbb{R}^{N\times N}$ denote the diagonal matrices:

$$D\left(\frac{\partial\phi_i}{\partial\hat{\alpha}_i}\right) = diag\left(\left(\frac{\partial\phi_1}{\partial\hat{\alpha}_1}\right), ..., \left(\frac{\partial\phi_N}{\partial\hat{\alpha}_N}\right)\right)$$

$$D\left(\frac{\partial\phi_i}{\partial\hat{p}_i}\right) = diag\left(\left(\frac{\partial\phi_1}{\partial\hat{p}_1}\right), ..., \left(\frac{\partial\phi_1}{\partial\hat{p}_N}\right)\right)$$

$$D\left(\frac{\partial\phi_i}{\partial\hat{p}_{i+N}}\right) = diag\left(\left(\frac{\partial\phi_1}{\partial\hat{p}_{N+1}}\right), ..., \left(\frac{\partial\phi_N}{\partial\hat{p}_{2N}}\right)\right)$$

whose diagonal elements are the partial derivatives of ϕ given by:

$$\left(\frac{\partial\phi_i}{\partial\hat{\alpha}_i}\right) = \frac{\hat{\alpha}_i}{\sqrt{(1-\hat{p}_i-\hat{p}_{i+N})^2+\hat{\alpha}_i^2}} - 1,$$

$$\left(\frac{\partial\phi_i}{\partial\hat{p}_i}\right) = -2\hat{p}_i\left(\frac{\hat{p}_i}{\sqrt{(1-\hat{p}_i-\hat{p}_{i+N})^2+\hat{\alpha}_i^2}} - 1\right),$$

$$\left(\frac{\partial\phi_i}{\partial\hat{p}_{i+N}}\right) = -2\hat{p}_{N+i}\left(\frac{\hat{p}_i}{\sqrt{(1-\hat{p}_i-\hat{p}_{i+N})^2+\hat{\alpha}_i^2}} - 1\right)$$

for all $1 \leq i \leq N$ if the Fischer-Burmeister function is differentiable hence $|\hat{p}|_i \neq 1$,$\hat{\alpha} \neq 0$ and consequently $\partial\Phi$ is a singelton.

In case of non-differentiabilility, hence if $|\hat{p}|_i = 1$ and $\hat{\alpha}_i = 0$, Ng et al. give the following subdifferentials:

$$\left(\frac{\partial\phi_i}{\partial\hat{\alpha}_i}\right) = \xi_i - 1, \left(\frac{\partial\phi_i}{\partial\hat{p}_i}\right) = -2p_i(\rho_i - 1), \left(\frac{\partial\phi_i}{\partial\hat{p}_{N+i}}\right) = -2p_{i+N}(\rho_i - 1)$$

with any scalars ξ_i and ρ_i for which holds: $||(\xi_i, \rho_i)^T|| \leq 1$. For a detailed derivation of these generalized differentials see [24].
The formal step direction in the Newton method for least squares is the following:

$$(V_k V_k^T)d = -V_k\Phi(z^k) \tag{3.81}$$

where $(V_k V_k^T)$ serves as the approximated Hessian and $V_k\Phi(z^k)$ as the gradient of Ψ. This quasi-Newton method is called Gauss-Newton method in literature and equation (3.81) is referred to as the normal equation. Far away from a KKT-point $(\hat{p}^*, \hat{\alpha}^*)$ the Hessian approximation of Ψ may not be accurate and the matrix $(V_k V_k^T)$ may not even be positive definite. A modification of the Gauss-Newton procedure can provide a remedy. In fact, the authors proposed the following modified Gauss-Newton equation to obtain the search direction:

$$(V_k V_k^T + \nu I)d = -V_k\Phi(z^k) \tag{3.82}$$

They add a positive constant ν on the diagonal of $(V_k V_k^T)$ in order to force positive definiteness. This constant should decrease approaching the solution vector $(\hat{p}^*, \hat{\alpha}^*)$. The

decrease rate should obviously be measured in terms of the natural merit function or its gradient. This is done by Ng in the following manner :

$$\nu = \min\{\Psi(z^k), ||\nabla\Psi||\} \tag{3.83}$$

Note that if ν is a large number, the eigenvalues of matrix $(V_k V_k^T + \nu I)$ are nearly uniform and the method performs similar to the gradient descent method. This can be interpreted as a fallback strategy, in case the Hessian approximation does not deliver a descent direction, by simply choosing the negative gradient as the descent direction. Near the minimum (small ν) the matrix $(V_k V_k^T + \nu I)$ provides a good approximation for the true Hessian and the quadratic convergence phase is likely to kick in. The gradient descent nature of this method in the damped Newton phase dictates a usage of a line-search procedure. This can be done by a simple backtracking algorithm as presented in section (2.3) using $V\Phi(z^k)$ as the gradient of Ψ. The complete algorithm can be listet as in algorithm (8). This modified

Algorithm 8 The modified Gauss-Newton method.

1. Choose $\rho \in (0,1)$,$\nu_0 > 0$, an initial point z^0 and set k:=0.
2. Choose a $V_k \in \partial\Phi(z^k)^T$ and solve the modified Gauss-Newton :
$$(V_k V_k^T + \nu I)d = -V_k\Phi(z^k)$$
3. Choose a step size t_k via backtracking line search that satisfies:
$$\Psi(z^k + t_k) \leq \Psi(z^k) + \rho t_k \nabla\Psi(z^k)^T d^k$$
4. Update z^k to z^{k+1}:
$$z^{k+1} := z^k + t_k d^k$$
5. Stop or choose $\nu_{k+1} = \min\{\Psi(z^k), ||\nabla\Psi||\}$, set and $k = k+1$ and go to (2).

Gauss-Newton algorithm is also referred to as the Levenberg-Marquardt method [42, 43]. The choice (3.83) is one of several strategies for choosing the parameter ν. In the same work the authors also propose an algorithm that solves a non-linear system of equations which only hinges on $\hat{p}$. This is done by taking for the Lagrange multiplier the analytic formulation given in equation (3.16).

Conclusion

In their work Ng et al. state that quadratic convergence occurs near the minimizer. However, many iterations (hundreds) are necessary to reach the quadratic phase of convergence. The difficulty of solving the pure dual formulation with Newton's method is that the divergence operator is rank deficient. This is due to its differential nature i.e all constant functions are in the null space of the ∇ operator. The dual formulation is therefore not strictly convex and the solution vector is possibly not unique. A possible remedy is to demand more from the dual variable p by introducing an additional penalty term. This can be done by a simple Tikhonov regularization [61]. In the next section, we derive the regularized dual problem and state its equivalence with a local smoothing of the primal problem. We also propose an alternative method for solving the underlying KKT system based on the so-called primal-dual active set method.

3.10 Our Proposed Method

In section (1.1.2) we studied a selection of measures of the gradient:

$$\int_\Omega \phi(\nabla u)\, dx \tag{3.84}$$

and observed that some are edge preserving and some are not. The choice of the Euclidean norm for ϕ as in the ROF model proved to be the best choice since it preserves edges and provides an isotropic diffusion within its gradient flow. Numerically, the L^1-norm poses the difficulty that it is not differentiable where $\nabla u = 0$. So one may ask whether there is an alternative measure which provides similar results and is smooth everywhere. It turns out that to denoise an image while respecting its discontinuities the function ϕ has to have *utmost linear growth at infinity* c.f. [15]. It is usual in analysis to approximate a given function by a more regular one. A popular approach is based on the so-called *infimal convolution.*

3.10.1 The Infimal Convolution

For illustration purposes we demonstrate the infimal convolution on the absolute function. The extension to the Euclidean norm in $\mathbb{R}^n$ is straightforward.

Definition 5 *Let f_1 and f_2 be convex and closed. The infimal convolution of f_1 with f_2 is given as:*

$$f(x) := (f_1 \star_{\inf} f_2) := \inf\{f_1(y) + f_2(x-y) : y \in \mathbb{R}^n\}.$$

The resulting function $f(x)$ is closed and convex since f_1 and f_2 are closed convex functions (see [36] Prop 2.1.3). We next consider a type of regularization based on the infimal convolution with a convex kernel K. If we set $K = \frac{1}{2\varepsilon}y^2$ we get the following infimal convolution:

$$H_\epsilon(x) := inf\left\{f_1(y) + \frac{1}{2\varepsilon}(x-y)^2 : y \in \mathbb{R}\right\}. \tag{3.85}$$

Setting $f_1 = |\cdot|$ gives us

$$H_\epsilon(x) := inf\left\{|y| + \frac{1}{2\varepsilon}(x-y)^2 : y \in \mathbb{R}\right\}. \tag{3.86}$$

This yields to the following regularization:

$$H_\epsilon(x) = \begin{cases} \frac{1}{2\varepsilon}x^2 & \text{for } |x| \le \varepsilon \\ |x| - \frac{\varepsilon}{2} & \text{else} \end{cases} \tag{3.87}$$

Function $H_\epsilon(x)$ is also known from robust statistics and is referred to as the Huber function [37]. The effect of the *Huber* regularization on the absolute function is illustrated in figure (3.3). Replacing f by $H_\epsilon(x)$ is also called *Moreau-Yosida* regularization. The regularization $H_\epsilon(x)$ is continuously differentiable (C^1) everywhere and shares the same minimizers with function f. Additionally, we observe that the Huber function exhibits linear growth for $|x| \to \infty$ from which we expect that edges will be only linearly penalized and hence remain existent. The analog for the Huber function in two dimensions is simply the following function:

$$H_\epsilon(\vec{q}) = \begin{cases} \frac{1}{2\varepsilon}|\vec{q}|^2 & \text{for } |\vec{q}| \le \varepsilon \\ |\vec{q}| - \frac{\varepsilon}{2} & \text{else} \end{cases} \tag{3.88}$$

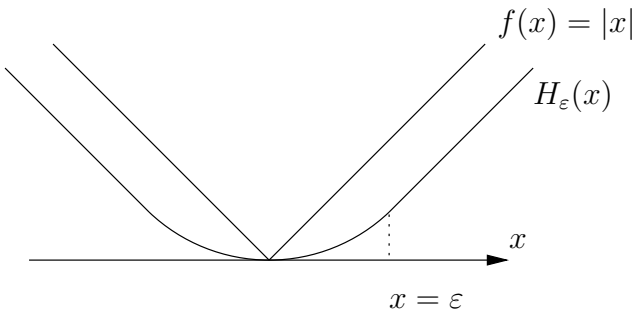


Figure 3.3: The Huber regulariaztion replaces the kink of the absolute function with a quadratic approximation. The choice of ϵ determines the transition from quadratic to linear energy.

Where $|\cdot|$ denotes the Euclidean norm. If we take the Huber function as a substitute for ϕ in equation (3.84), we obtain the so-called *Gauss-TV* functional.

3.10.2 The Gauss-TV Model

If we restate the unconstrained ROF model, incorporating the Huber penalty function, we get the following energy functional:

$$J_\epsilon(u) = \int_\Omega H_\epsilon(\nabla u)\, dx + \frac{1}{2\lambda}\int_\Omega (u-f)^2 dx. \tag{3.89}$$

Looking at the definition of the Huber function (3.88), we can write out equation (3.89) as:

$$J_\epsilon(u) = \frac{1}{2\epsilon}\int_{|\nabla u(x)|\leq\epsilon} |\nabla u|^2\, dx + \int_{|\nabla u(x)|>\epsilon} |\nabla u|\, dx + \frac{1}{2\lambda}\int_\Omega (u-f)^2 dx. \tag{3.90}$$

For regions with $|\nabla u(x)| < \epsilon$ the Sobolev semi-norm is used which performs a Gaussian-like smoothing. This regularizer does not preserve edges and does not have to in this context as edges exhibit a large gradient magnitude. For regions with $|\nabla u(x)| \geq \epsilon$ TV regularization is performed. This way ϵ acts as a switch between a Gaussian-like regularization for small $|\nabla u|$ and TV-regularization for large gradients. This type of regularization is also known as *Gauss-TV* [59] and was shown to be a good choice in medical imaging [39].

The total variation prior assumes that an ideal image consists of piecewise constant regions. This explains the tendency of TV to transform smooth regions into piecewise constant regions. The phenomena is called *staircasing* [7] and is a drawback of using TV as a regularizing energy.The Huber regularization deals with this case effectively by penalizing small gradient magnitudes quadratically. This makes the usage of the Huber regularization even more attractive. The staircasing effect will be showcased in the numerical results chapter. Furthermore we will show that the Huber regularization reduces staircasing effectively.

Note that although we eliminated the singularity in the ROF functional it is still highly nonlinear in TV regularized regions. In the following, we derive the dual problem of $J_\epsilon(u)$ with help of the Fenchel conjugate function. Priorly the Fenchel conjugate concept and its relation with the dual problem is described.

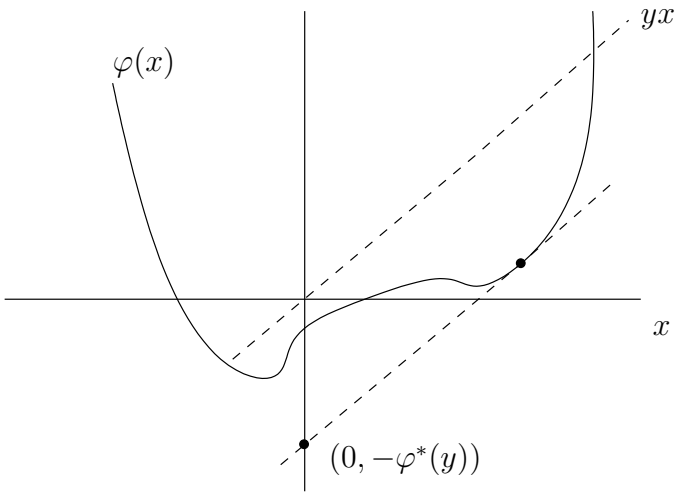


Figure 3.4: The conjugate function is the maximum gap between the linear function $\langle y, x\rangle$ and $\varphi(x)$. For a differentiable function this case occurs exactly at $(x, \varphi(x))$ where $\varphi'(x) = y$.

3.10.3 The Fenchel Conjugate Function

Convex conjugate functions are important for the investigation of optimization problems. The so-called dual optimization problem can be formulated by means of convex conjugate functions. This often allows a deeper understanding of the problem's structure. Often the dual problem is easier to solve than the primal counterpart. In what follows, we utilize the vector space V and its topological dual V^*. Additionally, the extended-value notation for the image of a convex function $f(x)$ is used. This is a standard notation in convex analysis which assigns $f(x) = \infty$ if $x \notin dom\ f$. The domain is recovered by setting $dom\ f = \{x | f(x) < \infty\}$.

Definition 6 *For a convex function $\varphi : V \to \mathbb{R} \cup \{\infty\}$ the Fenchel conjugate function $\varphi^* : V^* \to \mathbb{R} \cup \{\infty\}$ is defined on the dual space V^* and is given by:*

$$\varphi^*(y) = \sup_x \{\langle y, x\rangle - \varphi(x)\}.$$

Thus the conjugate function is the maximum gap between the linear functional $\langle y, x\rangle$ and $\varphi(x)$. Figure (3.4) illustrates the concept of conjugate functions in the one-dimensional case.

3.10.4 The Fenchel Dual Problem

Following the concept of duality, stated by Ekeland and Temam in [23], we derive the Fenchel dual of a optimization problem $(\mathcal{P})$ and denote it by $(\mathcal{P}^\star)$. Let $\Lambda \in \mathcal{L}(V, Y)$ i.e a bounded linear operator and let $F : V \to \mathbb{R} \cup \{\infty\}$ and $G :\to \mathbb{R} \cup \{\infty\}$ be convex functionals. Now suppose the primal problem can be decomposed into the form:

$$\inf_{u \in V} \{\ \mathcal{F}(u) + \mathcal{G}(\Lambda u)\} \qquad (\mathcal{P}).$$

According to Ekeland and Temam ([23] remark 4.2) the dual problem $(P^\star)$ can be written as:

$$\sup_{p \in Y^*} \{-\mathcal{F}^*(\Lambda^* p) - G^*(-p)\} \qquad (\mathcal{P}^\star)$$

where F^* and G^* are the Fenchel conjugates of F^* and G^*. Here Λ^* denotes the adjoint operator of Λ. Now we have the above duality framework, the Fenchel dual problem of the Gauss-TV problem can be derived (3.89) by making the following identifications:

$$\begin{aligned} \mathcal{F}(u) &:= \frac{1}{2\lambda}|u-f|^2 \\ \mathcal{G}(\nabla u) &:= \int_\Omega H_\epsilon(\nabla u) \end{aligned}$$

where $\Lambda := \nabla$. To finish the dual transform of problem $(\mathcal{P})$ we have to determine the Fenchel conjugate functions $\mathcal{F}^*$ and $\mathcal{G}^*$. As for the conjugate function of H_ϵ we resort to the infimal convolution formulation for which the following holds:

$$(f_1 \star_{inf} f_2)^* = f_1^* + f_2^*$$

hence the Fenchel conjugate of an infimal convolution is the sum of the conjugates of its arguments (see [36] Prop. 6.3.1). This gives us the Fenchel transform of the Huber function:

$$H_\epsilon(q)^* = (|q|)^* + (\frac{1}{2\epsilon}|q|^2)^*.$$

The Fenchel conjugate of the (ℓ_p) norm is the indicator function of the unit ball induced by the dual norm (ℓ_q) with $\frac{1}{p}+\frac{1}{q}=1$ (see e.g. [58] for a proof). Since the Euclidean norm is self-dual ($\frac{1}{p}+\frac{1}{q}=\frac{1}{2}+\frac{1}{2}=1$) we obtain the indicator function of the Euclidean unit ball. The conjugate of the Kernel $K=\frac{1}{2\epsilon}|q|^2$ can be obtained by applying the definition of the conjugate function:

$$K^*(p) = \sup_q \left\{ pq - \frac{1}{2\epsilon}|q|^2 \right\}.$$

We get rid of the supremum by setting the gradient $\nabla_q \left(pq - \frac{1}{2\epsilon}|q|^2\right) = 0$ and substituting the value for q in the equation to obtain:

$$K^* = \frac{\epsilon}{2}|p|^2$$

Finally, the conjugate of G looks as follows:

$$\mathcal{G}^*(p) = I_{(|p|\leq 1)} + \frac{\epsilon}{2}|p|^2. \tag{3.91}$$

with:

$$I_{(|p|\leq 1)} = \begin{cases} 0 & \text{if } |p(x)| \leq 1 \text{ for a.e. } x \in \Omega \\ \infty & \text{otherwise} \end{cases}.$$

As for the Fenchel conjugate of $\mathcal{F}$ we find:

$$\mathcal{F}^*(v) = \frac{1}{2\lambda}\left(|\lambda v + f|^2 - |f|^2\right).$$

Now we can apply the definition of the dual problem $(\mathcal{P}^*)$. We set $v = -\text{div } p$ knowing that $\Lambda^* = \nabla^* = -\text{div}$ and drop the additive constant $\frac{1}{2\lambda}|f|^2$ and the factor $\frac{1}{\lambda}$. The dual problem to (3.89) is then given by the constrained quadratic problem:

$$\inf_{|p|\leq 1} \frac{1}{2}|\lambda \text{div } p + f|^2 + \frac{\epsilon}{2}|p|^2. \tag{3.92}$$

Note that we changed the sign of p in order to stay consistent with the other dual formulations. This does not change the optimization problem since the unit disc is closed under reflection ($p \to -p$). The indicator function $I_{(|p|\leq 1)}$ gives us the constraint $|p| \leq 1$.

Interestingly, the Huber regularization of the primal ROF formulation coincides with a Tikhonov regularization of its Fenchel dual. This regularization is given by the term $\frac{\epsilon}{2}|p|^2$ and is a remedy to the non-uniqueness of the solution to the dual problem. Note that the fenchel dual problem for the functional (3.92) is again the Gauss-TV problem. This is because the Gauss-TV problem is a closed convex functional.

3.10.5 The Primal-Dual Active Set Method

Hintermüller and Kunisch (HK) consider in [41] the case of anisotropic TV. In their anisotropic formulation the Euclidean norm in the TV term is replaced by the (ℓ^1) norm. HK use the Fenchel dual approach to formulate a dual problem with a bilaterally constrained dual variable i.e. $-\vec{1} \leq p \leq \vec{1}$ where $\vec{1}$ denotes a vector with 1 in both coordinates. The authors pose the Lagrangian of the anisotropic dual ROF model as follows:

$$\min \frac{1}{2}|\lambda \mathrm{div} p + f|^2 + \frac{\epsilon}{2}|p|^2 + \frac{1}{2c}|\max(0, c(p - \vec{1}))|^2 + \frac{1}{2c}|\min(0, c(p + \vec{1}))|^2 \tag{3.93}$$

with $c > 0$. The minimization of the anisotropic TV is realized by a primal-dual active set (PDAS) method. We adapt this method in order to solve our optimization problem (3.92). By analogy with the Lagrangian formulation (3.93) we bring the constraints of problem (3.92) into equation:

$$F(p) = \min \frac{1}{2}|\lambda \mathrm{div} p + f|^2 + \frac{\epsilon}{2}|p|^2 + \frac{1}{2c}\max(0, c(|p| - 1))^2. \tag{3.94}$$

The penalty term $\frac{1}{2c}\max(0, c(|p| - 1))^2$ makes sure that the constraints are fulfilled at an optimal point p^*. Thus we differentiate the following two cases:

- If p is not feasible the maximum term is not zero and the objective penalizes the non-feasibility with a quadratic energy of $\frac{1}{2c}c(|p| - 1)^2$. This is done similar to the usage of the log barrier function in interior point methods although in our case the variable is allowed to leave the feasible set. In literature such function is called *external penalty* function.

- Otherwise, in case p stays in the unit circle, the penalty term does not contribute to the energy, hence is equal to zero.

The optimality condition of problem F (3.94) is the following system of equations:

$$\begin{aligned} g(p) &= -\lambda \nabla \mathrm{div} p^* - \nabla f + \epsilon p^* + \alpha_c = 0 \\ \text{with} \quad \alpha_c &= max\,(0, c\,|p^*| - 1)\,\frac{p^*}{|p^*|}. \end{aligned} \tag{3.95}$$

α_c can be interpreted as a Lagrange multiplier. It has a value of zero on inactive constraints and is > 0 on active constraints. Equation 3.95 is, on the contrary to the optimality of the anisotropic case, nonlinear due to the non-linear constraints. Additionally, we deal with a nonsmooth equation since the point-wise maximum operator is semismooth. We linearize equation $g(p) = 0$ by taking the derivative with respect to p and utilizing

the generalized derivative for the max-operator stated in equation (2.44). The resulting Hessian can be written as follows:

$$H(p) = -\lambda\nabla\mathrm{div} + \epsilon I + \left(c\frac{p}{|p|}\frac{p}{|p|}^T + c(|p|-1)\frac{I - \frac{p}{|p|}\frac{p}{|p|}^T}{|p|} \right)\chi_{\mathcal{A}}.$$

I stands for the two by two identity matrix and $\chi_{\mathcal{A}}$ denotes the characteristic function of the set of variables p that violates the constraint, i.e where the maximum term in equation (3.95) is not zero. This set is called the active set and we denote it by $\mathcal{A}$. The characteristic function $\chi_{\mathcal{A}}$ takes the following values:

$$\chi_{\mathcal{A}} = \begin{cases} 1 & \text{if } x \in \mathcal{A} \\ 0 & \text{if } x \in \mathcal{I} \end{cases}$$

with index set $\mathcal{A}$ and the inactive index set $\mathcal{I}$ defined as:

$$\begin{aligned} \mathcal{A} &= \{x : (|p(x)| - 1) > 0\}, \\ \mathcal{I} &= \Omega \backslash \mathcal{A}. \end{aligned}$$

In algorithm (9), we propose an active set method for minimizing equation (3.94). The

Algorithm 9 The Primal Dual Active Set method

1. Choose $p_0 = 0$, $c > 0$, $\epsilon > 0$ and set $k = 0$
2. Determine the sets $\mathcal{A}$ and $\mathcal{I}$

$$\begin{aligned} \mathcal{A}_{k+1} &= \{x : (|p_k(x)| - 1) > 0\}, \\ \mathcal{I}_{k+1} &= \Omega \backslash \mathcal{A}_{k+1}. \end{aligned}$$

3. Solve for δp the follwoing system of equations:

$$H(p_k)\delta p_k = -g(p_k)$$

4. Update $p_{k+1} = p_k + \delta p_k$.
5. Stop, or set $k = k + 1$ and go to (2)

determination of the active set is the main objective of this algorithm. At every iteration the current iterate for p_k is used to predict the active set $\mathcal{A}_{k+1}$ and the inactive set for the next iteration. This means that we compute an estimate for the indices of those variables, for which we believe that $|p| = 1$ at the solution. Step 3 then solves the Newton system, based on the prediction made for this active set. Note that update step takes full step lengths of one. In experiments no step length restriction via line search was necessary for the algorithm to converge. The algorithm exhibits global convergence, hence independent of the initialization of p.

3.10.6 Convergence

In their work [34] Ito and Kunisch show that the semismooth Newton method converges locally superlinearly. The superlinear speed of convergence of our method is confirmed in experiments and will be illustrated in the numerical results chapter.

3.11 Conclusion

In this chapter we reviewed several different algorithms to solve the ROF model. It was shown that the naive application of Newton's method on the primal formulation of ROF leads to bad convergence results. This is due to the highly nonlinear Euler-Lagrange equation, for which the local quadratic approximation, which is the basis for the Newton method, is not a good model. We then discussed primal-dual Newton methods. These methods introduce a flux term in order to get get rid of the singularity in the Euler-Lagrange equation. Primal-dual methods solve essentially a saddle point problem and provide a very efficient way to minimize the ROF functional. However a perturbation of the optimality system is still necessary which prevent these algorithms to convergence to the true minimizer of the energy functional. Next we considered methods that solve the dual formulation of the ROF model. The energy functional of the this model exhibits a nice quadratic objective and needs not to be perturbed. However, one have to deal with many points-wise constraints that need sophisticated optimization algorithms. We studied two different philosophies for solving its underlying KKT system:

- The first approach is based on the so called Fischer-Burmeister function, which handles the complementary assumptions in the KKT optimality conditions. This approach is inefficient due to the rank deficient divergence operator which make the dual problem not strictly convex.

- As a second approach, we have proposed an active set Newton method to solve the dual formulation of the ROF model. The dual objective is regularized in the sense of Tikhonov and we showed that this regularization corresponds to a local smoothing of the primal formulation of ROF, and is a remedy to the non-uniqueness of the solution to the original dual problem. As a nice by-product, the applied H^1 regularization also helps reducing the staircasing effect.

Chapter 4

Numerical Results

In this chapter we present some numerical results, obtained from the Newton methods discussed in this work. Firstly, the implementation aspects of these methods is addressed. As a second step, we showcase the performance of our proposed algorithm on image denoising and show that the staircasing effect is reduced, compared to conventional ROF denoising. Then, a comparison of the Newton methods presented in the last chapter including our proposed method, is carried out. This comparison will include performance evaluations as well as visual comparison of denoised images. Finally, we demonstrate image segmentation, by means of the ROF model, which is described in the first chapter, by using the proposed PDAS algorithm.

4.1 Implementation

During this thesis the following algorithms for ROF minimization were implemented:

- The primal-dual algorithm of Hintermüller and Stadler.
- The semismooth algorithm of Ng et al.
- The proposed primal-dual active set method.

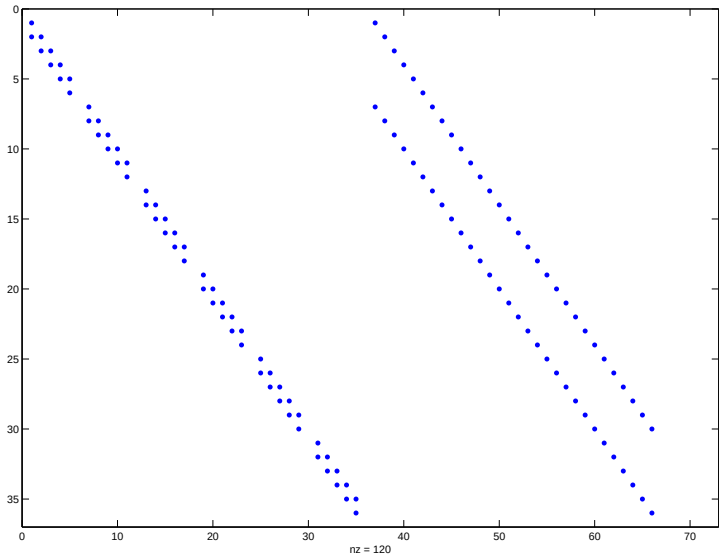
All methods are implemented in pure MATLAB on a 2.2 Ghz Intel Core Duo CPU and 4GB memory. It is expected that the performance of the algorithms would improve if implemented in $C++$ and we expect the speedup to be uniform for all algorithms. The code of the CGM method was downloaded from the web page of Stephen J. Wright[1]. Throughout all methods we make use of the matrix-vector representation from section (3.2.3). Hence, the spacial difference operations are represented by the matrices:

$$\begin{aligned} A^T &= \nabla \\ -A &= \text{div}. \end{aligned}$$

4.1.1 Storing sparse Matrices

The main difficulty of implementing the Newton algorithms is that the linear systems, solved in every iteration, are of huge size. For example a 128×128 pixel image, solved with our proposed method, generates a 32768×32768 matrix for the Hessian. Storing these matrices explicitly is therefore often not possible due to their dimensions. However,

[1] http://pages.cs.wisc.edu/ swright/TVdenoising/



(a)

Figure 4.1: The sparsity pattern of A for an 6×6 image. The blue dots represents non-zero elements. The matrix has the dimensions 36×72 and possesses only 120 non-zero elements

these matrices are extremely sparse and can be stored using elaborate data structures. The sparsity pattern of matrix A for an 6×6 image is illustrated in figure (4.1). MATLAB provides data structures for sparse matrices. By following the construction rules in section (3.2.3), matrix A can be constructed by means of the `spdiags` command in MATLAB. This command generates sparse diagonal matrices and requires only the specification of the diagonal vectors as an argument. The following MATLAB code summarizes the construction of the differential matrix A:

```
e1=ones(n, 1);
A1=spdiags([e1 -e1], [-1 0], n, n);
A1(n, n)=0;
A=[ kron(speye(n), A1) kron(A1, speye(n))];
```

The image is again assumed to be of size $n \times n$. The matrix `A1` is the one-dimensional backward difference ∇_1 and is constructed by taking the diagonals `e1` and `-e1` as arguments. The vector `e1` is the n-dimensional vector consisting only of ones in every coordinate. The second argument of `spdiags` is the vector `d=[-1,0]` which does nothing else than specifying the diagonals to be filled. The diagonals of a matrix can be specified by a number k where:

$$j = i + k.$$

Thus, the entry 0 stands for the main diagonal and -1 for the sub-diagonal. We set `A1(n,n)=0` to ensure that boundary conditions hold for the forward difference operator ∇_1. The last call is exactly the formula:

$$A = \begin{bmatrix} I_n \otimes \nabla_1 & \nabla_1 \otimes I_n \end{bmatrix}$$

from section (3.2.3). The matrix `speye(n)` is a sparse implementation of the $n \times n$ identity matrix I_n.

Finally, the maps Π_N and Π_{2N} are realized with a reordering command in MATLAB, called `reshape`.

4.1.2 Solving the Linear System of Equations

The linear systems of equations that occur in every iteration, are solved with help of the *backslash* command in MATLAB. The Newton linearized system is solved as follows:

```
dx=H\-G;
```

`H` respesents the Hessian matrix and `G` denotes the gradient. The above command returns the step denoted by `dx`.This direct solver is a collection of algorithms that decides, based on the sparsity pattern of the Hessian, which algorithm to use in order to solve the linear system of equations. As discussed in the last chapter, the Hessian in all the algorithms is positive definite, either by default or after a modification. According to MATLAB's reference of the backslash operator, it performs a sparse Cholesky factorization if the matrix is symmetric positive definite. The Cholesky factorization is a common technique for solving matrices in S_{++} and gives for the Hessian `H` the following factors:

$$H = LL^T$$

L is a lower triangular matrix with strictly positive diagonal entries and L^T is the transposed matrix. The solution for the system of equations is then obtained by solving first $Ly = -G$ for y and finally $L^T dx = y$ for the Newton step dx, which is easily done by forward and backward substitution.

We also tried solving the system of equations iteratively with the conjugate gradient method. However, the convergence of the algorithm was significantly slower in that case.

4.2 The PDAS algorithm

In this section the performance of the PDAS algorithm, proposed in the last chapter, is demonstrated. We illustrate the denoising capabilities of the PDAS algorithm on three test problems, consisting of noisy 256×256 images. The images take values in the intervall $[0, 255]$ and are artificially degraded with a Gaussian noise of standard deviation of $\sigma = 0.05$ and a mean value of 0. The noisy images and the clean originals are confronted in figure (4.2). As mentioned in section (1.2.1), the fidelity parameter λ should be chosen proportionally to the noise level. We found that $\lambda = 15$ is adequate for this level of noise. Additionally, we choose $\epsilon = 0.1$ and we find that setting $c = 100$ is sufficient for our purpose. Using these parameters, image denoising on our test problems, is performed.

The algorithm terminates as soon as the residual $||g(p)||$ drops below 10^{-6}. Clearly, the denoised images still have sharp edges as expected.

4.2.1 Staircasing

As mentioned in the last chapter the Huber regularization helps reducing the so-called staircasing effect. This is done by penalizing smooth regions quadratically instead of using TV regularization. The effect of the Huber regularization on smooth regions is displayed in figure (4.4) where our method is compared to the result of the CGM algorithm. Obviously, the CGM algorithm tends to produce piecewise constant regions in smooth parts of the image. From the the depicted signals, one sees that the CGM method produces plateaus in

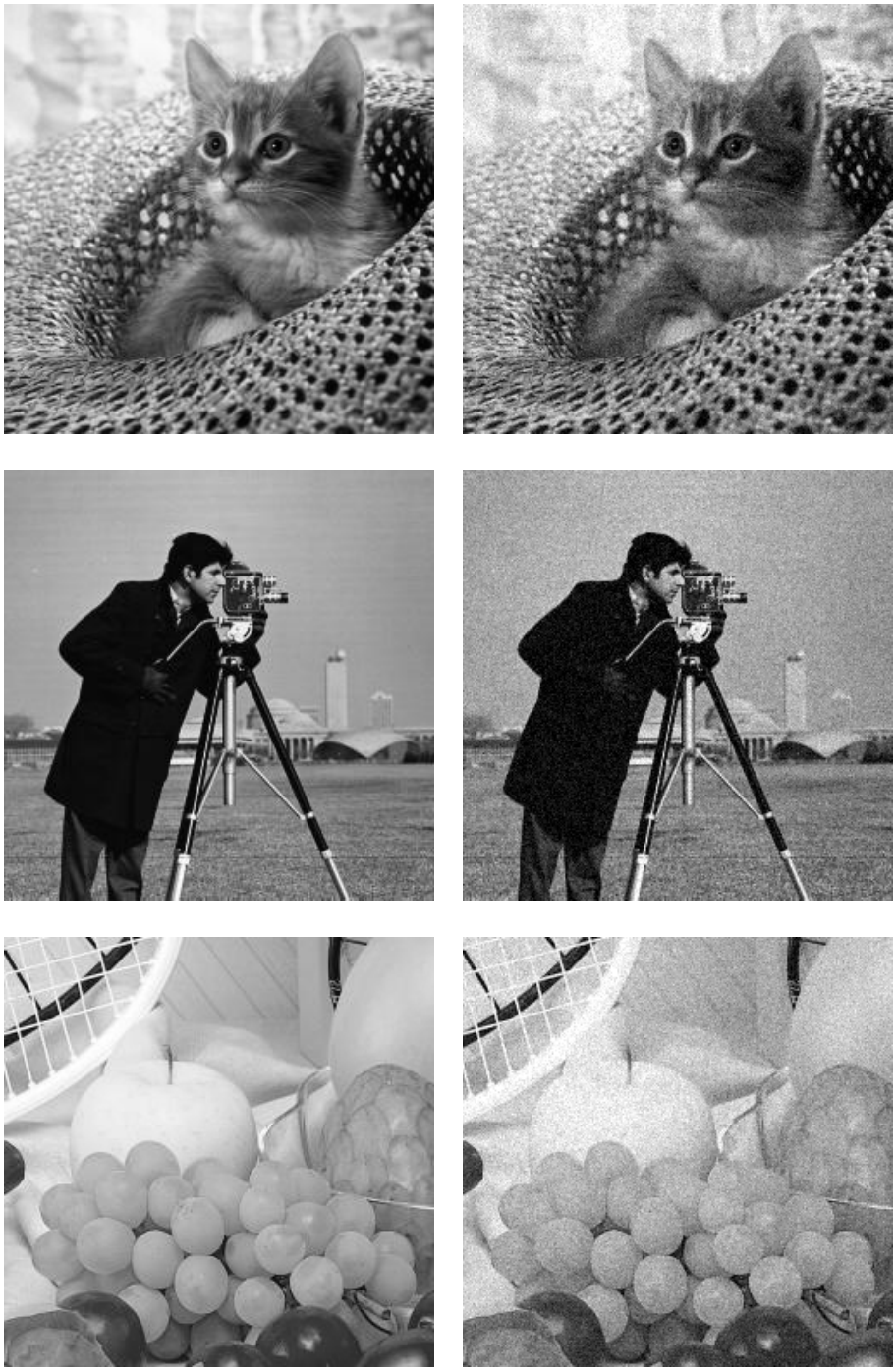


Figure 4.2: The clean images and their degraded versions with a noise of $\sigma = 0.05$ standard deviation and a mean value of zero.

Figure 4.3: The denosing results with the PDAS algorithm for $\lambda = 15$.

the reconstruction. In contrast, our method reduces the staircasing effect and the resulting ramp does not make any jumps like in the CGM result. For the smoothing parameter we used $\lambda = 40$ and we set for the Huber threshold $\epsilon = 0.1$.

4.3 Comparision

We will now compare the CGM, HS, Ng and our proposed PDAS algorithm with regard to visual results and convergence properties. All convergence plots will plot the iteration number against the logarithmic scaled natural residual of the corresponding optimality equation.

Figure (4.5) shows a comparison of denoising results from all Newton algorithms. As mentioned in the last chapter the semismooth algorithm of Ng et al. exhibits first-order convergence far away from the optimum and is therefore very slow. Figure (4.6) shows a convergence plot of the Ng algorithm compared to the other Newton methods for $\lambda = 15$. One observes that the Ng algorithm needs more than 250 iterations while the other methods converge within the first 20 iterations. Therefore, we exclude the Ng algorithm from further convergence results as it is not nearly competitive with the other methods.

Next, a comparison of the convergence of the HS, CGM and PDAS algorithm is carried out. Figure (4.7) shows convergence plots of the algorithms with different values for λ. Note that all algorithms converge, to machine precision, within the first 20 iterations. However, the HS algorithm outperforms the other algorithms asymptotically.

The CGM algorithm in this implementation is prone to small values for β and exhibits the slowest convergence. But we expect the modified CGM method, presented by Zhu in [67], to converge faster.

Observe that the PDAS as well as the HS algorithm are not monotone methods, as the residual often increases which is typical for semi-smooth Newton methods. These algorithms are therefore not descent methods.

Table (4.1) reports the result of a performance evaluation of the PDAS, CGM and HS made for different values for λ. The table entries refer to the number of iterations necessary, to attain an accuracy of 10^{-6} and the execution times in seconds. Overall, the plots and the table show that for larger values of λ the iteration number, as well as the execution times, increase. This is expected, as for a large λ the TV regularization is more pronounced and the ROF problem gets harder to solve.

Note that although the HS algorithm requires the least of iterations, it exhibits high execution times, particularly when λ is large. In contrast, the execution times of the CGM and PDAS method scale moderately with an increasing value for λ.

λ	15	20	50	70
CGM	32.7/17	37.8/16	40.5/17	45.8/21
HS	27/11	36.8/12	54.9/20	58.1/20
PDAS	27/17	25.8/15	36.1/27	49.5/37

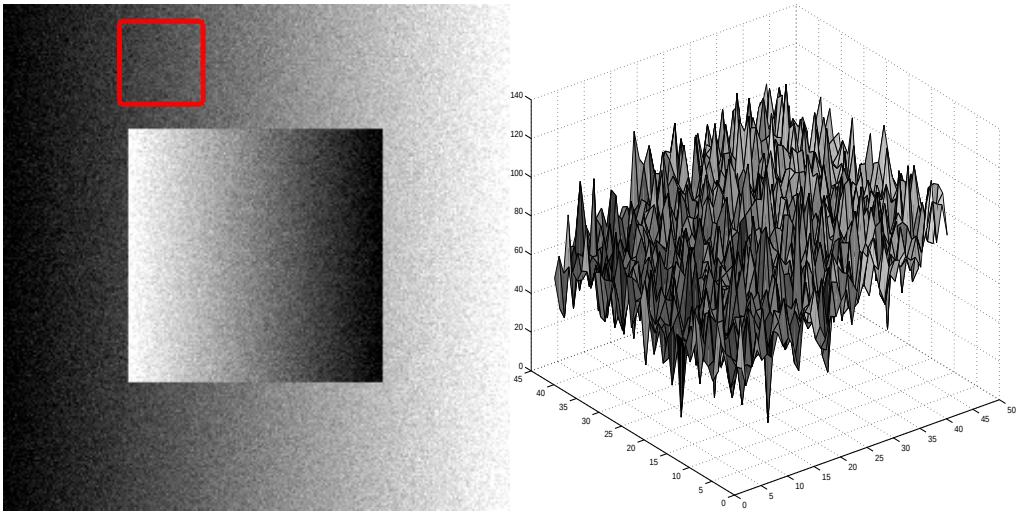
Table 4.1: Performance of the CGM, HS and PDAS algorithm measured in execution time and iteration number (time in seconds/iterations).

4.4 Segmentation via the ROF model

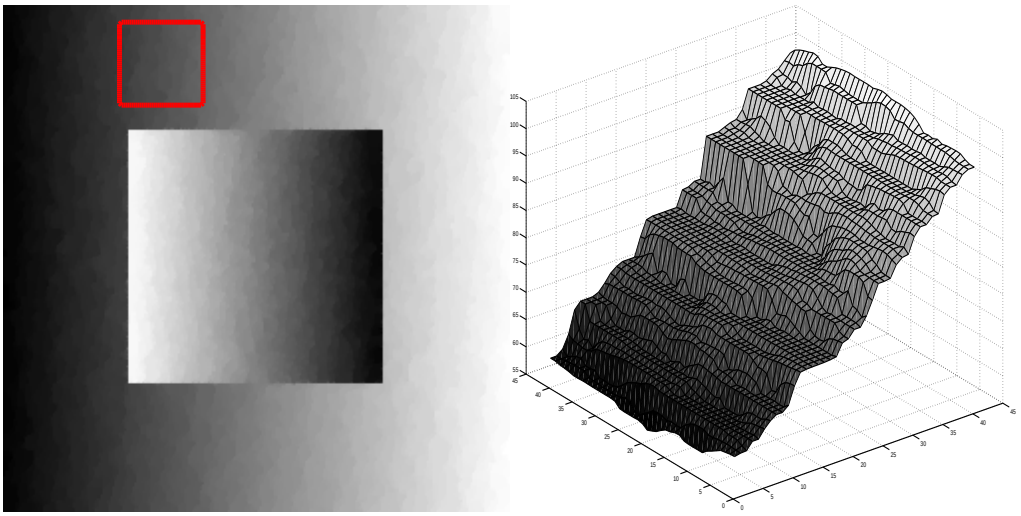
In section (1.3.7) we showed how to solve the two-phase Mumford-Shah functional by solving a modified ROF problem and taking the zero upper-level set of the solution. In this section, we demonstrate the segmentation capabilities of the ROF model by using the proposed PDAS algorithm. For the constants c_1 and c_2 we choose typical intensity values of the object, and respectively the background. These values are apparent in the histograms of the images. As discussed, for the input data we take:

$$\tilde{f} = (c_2 - f)^2 - (c_1 - f)^2$$

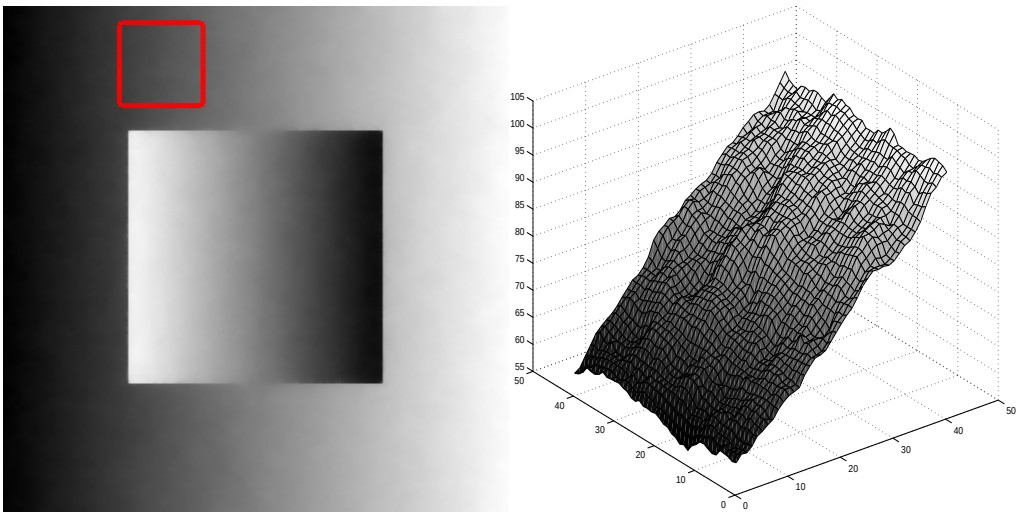
instead of the input image f as in the conventional ROF model. Note that the image f is scaled to the interval $[0, 1]$ this time, leading to smaller equivalent values for λ. After obtaining the solution u^* of the modified ROF model, the binary function $\{u^* > 0\}$ is the desired segmentation. Figure (4.8) shows the ROF segmentation procedure applied on the cameraman image. Figure (4.9) illustrates the effect of different values for λ on the segmentation results. One observes that, if λ is chosen large, the perimeter of the segmented object is stronger penalized and the segmentation tends to avoid elongated structures like the branches of the tree. Figure (4.10) shows segmentation results for different images. Image (f) shows the limit of what is feasible with intensity based segmentation as not only the skier is segmented, but also other regions with similar intensities.



(a) Noisy ramp image (b) Noisy ramp signal

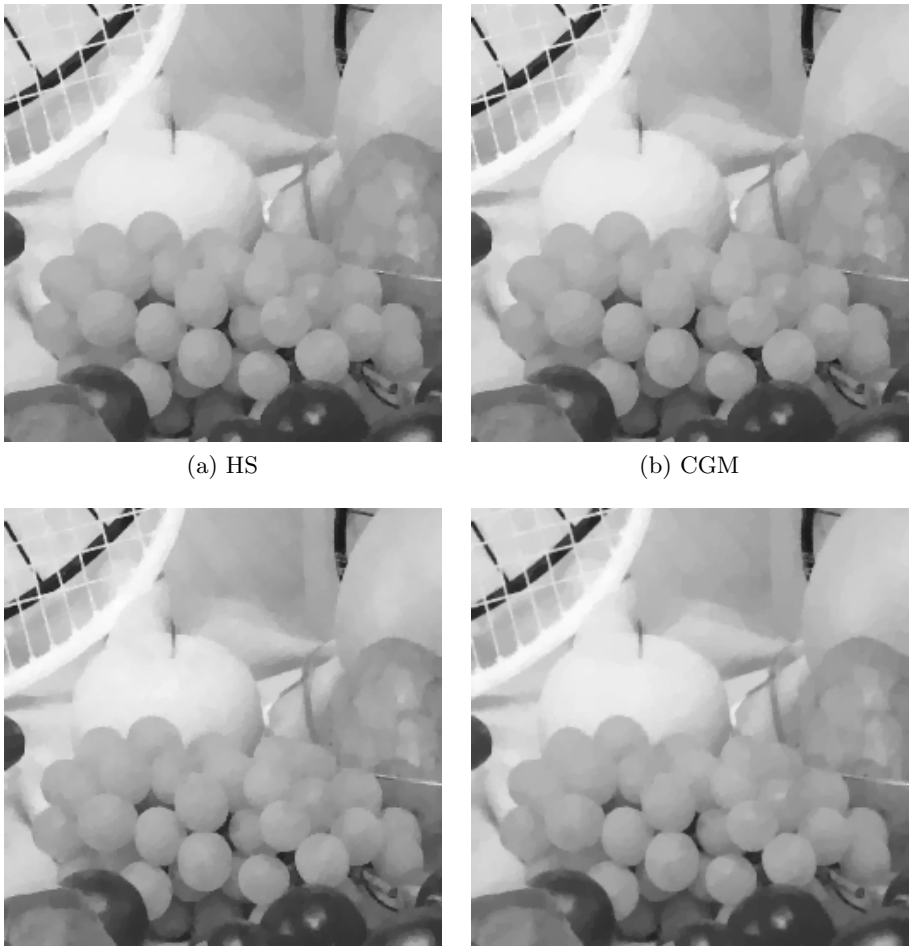


(c) CGM denoising (d) CGM denoised Signal



(e) PDAS denoising (f) PDAS denoised signal

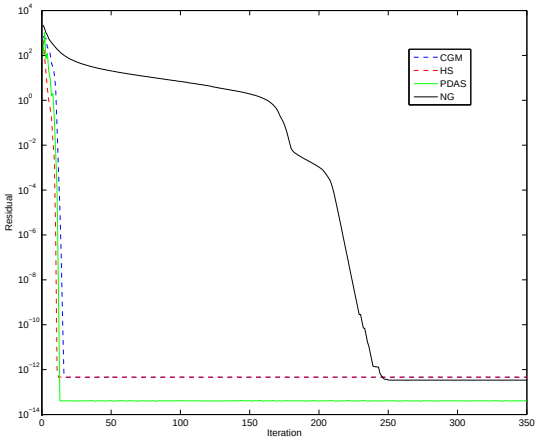
Figure 4.4: Compared to the CGM method, our proposed method reduces staircasing in the reconstructed image. The right column shows signal plots of selected areas (red rectangles) in the images on the left column.



(a) HS (b) CGM

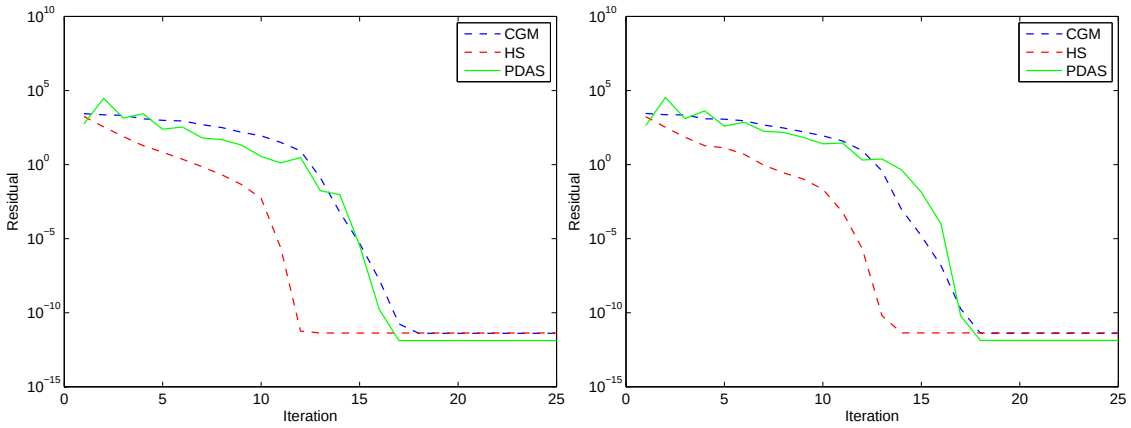
(c) PDAS (d) NG

Figure 4.5: Denoising results of the HS, CGM, PDAS and Ng algorithm with $\lambda = 15$.



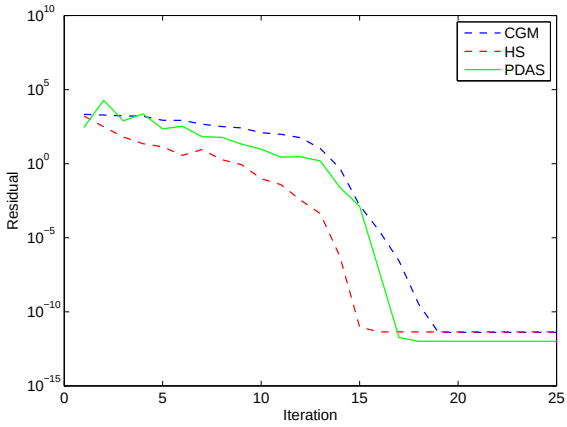
(a) Comparison of all algorithms over 350 iterations with $\lambda = 15$.

Figure 4.6: The convergence plot of the Ng algorithm compared to the CGM, HS and PDAS algorithm.

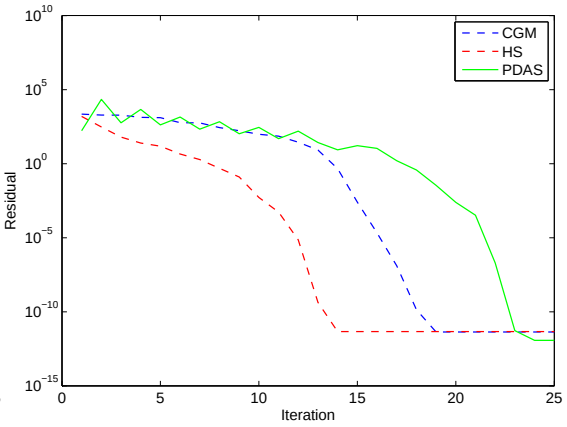


(a) $\lambda = 15$

(b) $\lambda = 20$

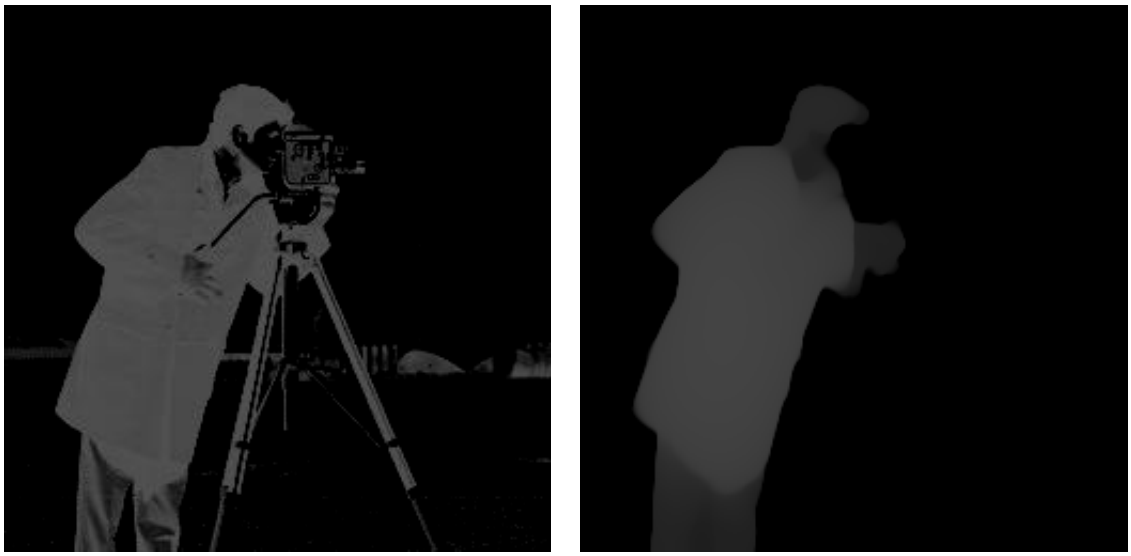


(c) $\lambda = 30$



(d) $\lambda = 50$

Figure 4.7: Convergence plots of the HS, CGM and the PDAS algorithm for different values for λ.



(a) $\tilde{f}$ (b) u^*

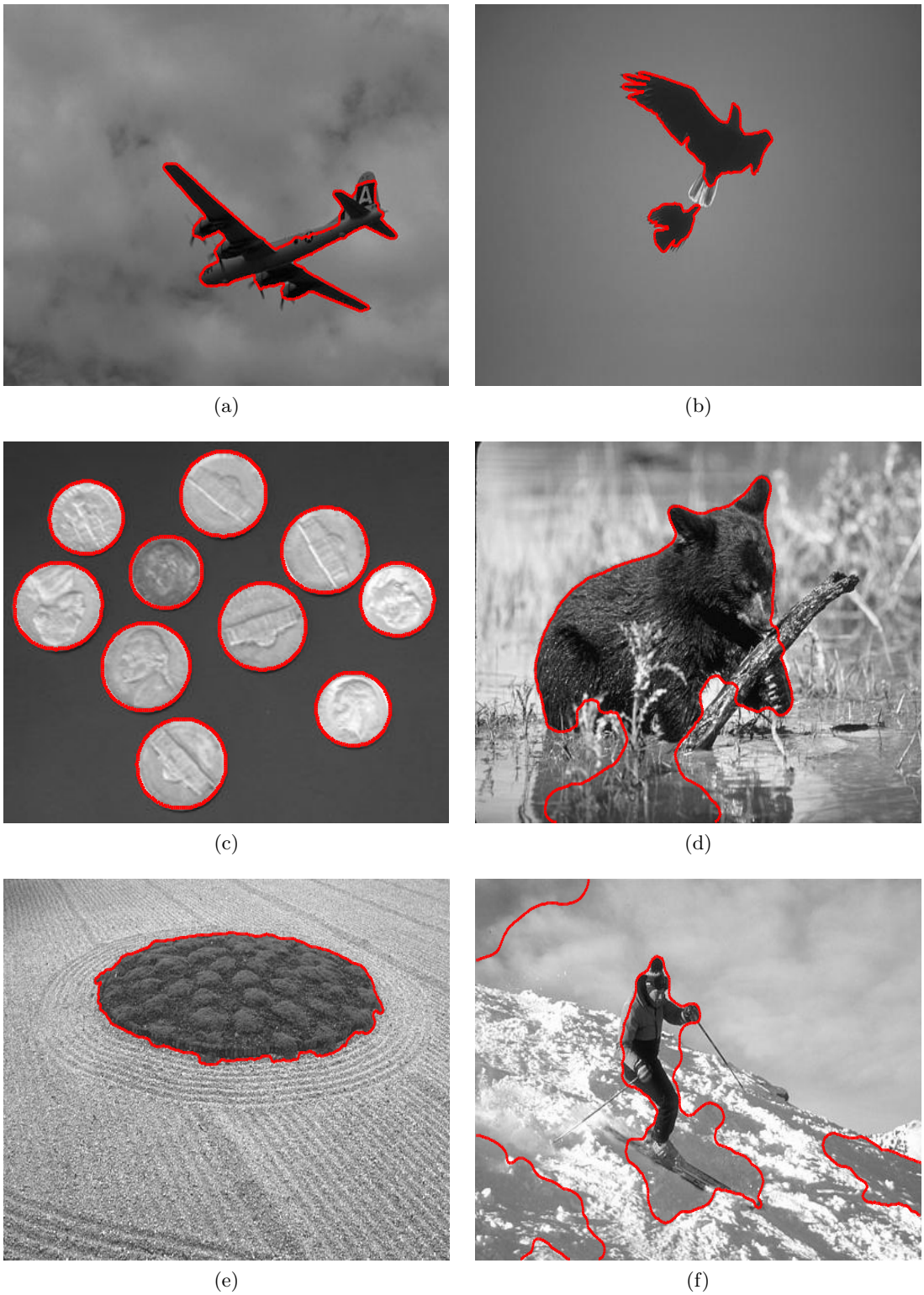
(c) $E = \{u^* > 0\}$ (d) contours of E

Figure 4.8: Image (a) is the data term $\tilde{f}$. Image (b) is the ROF minimizer u^*. Image (c) is the thresholded solution $\{u^* > 0\}$. Image (d) is the contours of the segmentation.

(a) $\lambda = 0.1$ (b) $\lambda = 1$

(c) $\lambda = 2$ (d) $\lambda = 3$

Figure 4.9: Segmentation results for different values for λ. With increasing λ the perimeter of the contur is stronger penalizes, leaving out elongated structures i.e. the branches in the above scene. This image is taken from [45].



(a) (b) (c) (d) (e) (f)

Figure 4.10: Segmentation results for different images. Image (f) shows the disadvantage of segmentation solely based on image intensity, as apart from the skier, other regions of the image are also segmented . The images are taken from the Berkeley segmentation dataset [45].

Chapter 5

Discussion and Future Work

In this work we proposed a novel method for solving the dual formulation of a Huber regularized ROF model. The numerical results from the last chapter show that our approach is highly competitive with the other Newton algorithms. In fact, and to the best of our knowledge, the above presented PDAS algorithm is the fastest Newton algorithm for solving the dual formulation of the ROF model. Additionally, and in contrast to the primal-dual approaches, only the dual variable has to be stored. This leads to a lower memory requirement of our method. Moreover we showed the advantage of the Huber-ROF approach, over purely TV regularized methods. That is, our method does not produce staircasing in the reconstruction results. This is due to the adaptive nature of the Huber regularizer which switches to a Sobolev regularizer in smooth regions.

The numerical results also show that all Newton methods exhibit high execution times, that are far from realtime performance. This results from the large linear systems that have to be solved in every iteration. Thus Newton's algorithms practically solve a quadratic sub-optimization problem in every iteration . This fact represents the major bottleneck of second-order optimization problems. Hence, the major challenge in accelerating Newton's method is of linear algebraic nature and can be topically divided as follows:

- **Parallel implementation**: Since computer architectures are becoming increasingly parallel, and the GPUs are the leading technology, it is natural to study parallel GPU implementations of algorithms for solving linear systems of equations. This represents an interesting challenge particularly when data structures for sparsity are necessary in order to store the system of equations, as it is the case for the Newton methods discussed in this work. In fact there is a lot of ongoing research in this matter c.f. [28, 32, 64].

- **Preconditioning techniques**: Another acceleration could provide sophisticated preconditioning techniques, since the solved linear systems are severely ill-conditioned. Especially when using the conjugate gradient method for solving the linear system, a preconditioning of the coefficient matrix is often necessary.

- **Investigating sparsity pattern**: The last chapter showed that the Newton linearized systems have a very sparse structure. The cost of the Cholesky factorization of Hessian depends heavily on the sparsity pattern of the matrix. Moreover, the sparsity of the the Hessian does not change as the variable varies. Thus it would pay off to devise a tailored strategy for solving the Newton system by exploiting these sparsity pattern. In fact in practice the complexity of the factorization can

be reduced from cubic to linear complexity. For more information see ([58] section 9.7).

However, considering these approaches is beyond the scope of this thesis. Though, it would be interesting to investigate the performance of the Newton algorithms for solving the ROF model, when implemented on parallel GPU architecture. These are remaining problems that would have to be addressed in future work. Until then, optimized first-order methods, e.g in the sense of Nesterov, remain the most efficient way of solving the ROF problem. Parallel implementation of these methods allow for even real time performance, in many computer vision tasks. Yet we believe that, in the long run and due to their superior asymptotic convergence, second-order methods will overtake gradient based methods, in terms of performance.

Appendix A

The Schur Complement Method

Often linear systems of equations posses a block structure. Consider the following system of equations with a $(p+q) \times (p+q)$ block coefficient matrix:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} \tag{A.1}$$

where $x \in \mathbb{R}^p$, $y \in \mathbb{R}^q$, $A \in \mathbb{R}^{p \times p}$, $B \in \mathbb{R}^{p \times q}$, $C \in \mathbb{R}^{q \times p}$, and $D \in \mathbb{R}^{q \times q}$. The so-called Schur complement of block D is the $p \times p$ matrix:

$$S = A - BD^{-1}C.$$

If we assume D to be invertible, the solution for x can be obtained by solving the following equation for x:

$$Sx = a - BD^{-1}b.$$

If we then substitute the value for x in the lower equation of system (A.1) of equations, hence in :

$$Cx + Dy = b$$

we obtain the following system of equations for y:

$$y = D^{-1}(b - Cx).$$

This so-called block elimination allows us to solve, instead of a $(p+q) \times (p+q)$ system, a smaller system of size $(p \times p)$ and respectively $(q \times q)$. Sometimes matrix D is of diagonal structure. In this case, calculating the inverse is trivial since D^{-1} is simply the diagonal matrix with the reciprocals of the diagonal entries of D. Solving system (A.1) then amounts to solving only a $(p \times p)$ system.

Bibliography

[1] S. Agmon. The relaxation method for linear inequalities. *Canadian Journal of Mathematics*, 6(3):382–392, 1954.

[2] Thomas Pock Antonin Chambolle. A first-order primal-dual algorithm for convex problems with applications to imaging. *preprint*.

[3] Kenneth J Arrow, Leonid Hurwicz, and Hirofumi Uzawa. *Studies in linear and nonlinear programming*. Stanford Math. Stud. Social Sci. Cambridge Univ. Press, Stanford, CA, 1958.

[4] Alfred Auslender and Marc Teboulle. Interior gradient and proximal methods for convex and conic optimization. *SIAM Journal on Optimization*, 16(3):697–725, 2006.

[5] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIIMS*, 2(1):183–202, 2009.

[6] Amir Beck and Marc Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *Trans. Img. Proc.*, 18(11):2419–2434, 2009.

[7] Peter Blomgren, Tony F. Chan, Pep Mulet, and C.K. Wong. Total variation image restoration: Numerical methods and extensions. In *in Proc. IEEE ICIP*, pages 384–387, 1997.

[8] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, 2004.

[9] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:2001, 1999.

[10] J. L. Carter. *Dual method for total variation-based image restoration*. PhD thesis, 2002.

[11] A. Chambolle, V. Caselles, M. Novaga, D. Cremers, and T. Pock. An introduction to Total Variation for Image Analysis, 2009.

[12] Antonin Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1-2):89–97, 2004.

[13] Antonin Chambolle. Total variation minimization and a class of binary MRF models. 3757:136–152, 2005.

[14] Antonin Chambolle and Jérôme Darbon. On total variation minimization and surface evolution using parametric maximum flows. *Int. J. Comput. Vision*, 84(3):288–307, 2009.

[15] Antonin Chambolle and Pierre-Louis Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2):167–188, April 1997.

[16] T.F. Chan and L.A. Vese. Active contours without edges. *IP*, 10(2):266–277, February 2001.

[17] Tony Chan and Jianhong Shen. *Image Processing And Analysis: Variational, Pde, Wavelet, And Stochastic Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2005.

[18] Tony F. Chan, Selim Esedoglu, and Mila Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. Technical report, SIAM Journal on Applied Mathematics, 2004.

[19] Tony F. Chan, Gene H. Golub, and Pep Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM J. Sci. Comput*, 20:1964–1977, 1999.

[20] Tony F. Chan, H. M. Zhou, and Raymond H. Chan. Continuation method for total variation denoising problems. Technical report, 1995.

[21] Xiaojun Chen, Zuhair Nashed, and Liqun Qi. Smoothing methods and semismooth methods for nondifferentiable operator equations. *SIAM J. Numer. Anal*, 38:2000, 1999.

[22] F.H. Clarke. *Optimization and Nonsmooth Analysis*. Wiley New York, 1983.

[23] I. Ekeland and R. Temam. *Analyse convexe et problèmes variationnels*. 1974. Collection Études Mathématiques.

[24] Francisco Facchinei, Andreas Fischer, Christian Kanzow, F. Facchinei, A. Fischer, and C. Kanzow. A semismooth newton method for variational inequalities: Theoretical results and preliminary numerical experience, 1997.

[25] A. Fischer. A special newton-type optimization method. *Optimization*, 24:269–284, 269-284.

[26] Wendell H. Fleming and Raymond Rishel. An integral formula for total gradient variation. *Archiv der Mathematik*, 11(1):218–222, December 1960.

[27] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *Computer Journal*, 7:149–154, 1964.

[28] Michael Garland. Sparse matrix computations on manycore gpu's. In *DAC '08: Proceedings of the 45th annual Design Automation Conference*, pages 2–6, New York, NY, USA, 2008. ACM.

[29] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. pages 452–472, 1990.

[30] Enrico. Giusti. *Minimal surfaces and functions of bounded variation / [by] Enrico Giusti ; notes by Graham H. Williams*. Australian National University, Canberra :, 1977.

[31] Donald Goldfarb and Wotao Yin. Second-order cone programming methods for total variation-based image restoration. *SIAM J. Sci. Comput*, 27:622–645, 2004.

[32] Gundolf Haase, Manfred Liebmann, , Craig C. Douglas, and Gernot Plank. A parallel algebraic multigrid solver on graphics processing units. In *High Performance Computing and Applications*, volume 5938 of *Lecture Notes in Computer Science*, pages 38–47. Springer Berlin / Heidelberg, 2010.

[33] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, Dec 1952.

[34] M. Hintermüller, K. Ito, and K. Kunisch. The primal-dual active set strategy as a semismooth newton method. *SIAM J. on Optimization*, 13(3):865–888, 2002.

[35] Michael Hintermüller and Georg Stadler. An infeasible primal-dual algorithm for total bounded variation-based inf-convolution-type image restoration. *SIAM J. Scientific Computing*, 28(1):1–23, 2006.

[36] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms I.* Springer-Verlag, Berlin, 1993.

[37] Peter J. Huber. Robust regression: Asymptotics, conjectures and monte carlo. *The Annals of Statistics*, 1(5):799–821, 1973.

[38] William Karush. Minima of functions of several variables with inequalities as side conditions. Master's thesis, Department of Mathematics, University of Chicago, Chicago, IL, USA, 1939.

[39] Stephen L. Keeling. Total variation based convex filters for medical imaging. *Appl. Math. Comput.*, 139(1):101–119, 2003.

[40] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized-via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:147–159, 2004.

[41] K. Kunisch and M. Hintermüller. Total bounded variation regularization as a bilaterally constrained optimization problem. *SIAM Journal on Applied Mathematics*, 64(4):1311–1333, 2004.

[42] K. Levenberg. A method for the solution of certain problems in least squares. *Quart. Applied Math.*, 2:164–168, 1944.

[43] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.

[44] Antonio Marquina and Stanley Osher. Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal. *SIAM J. Sci. Comput*, 22:387–405, 1999.

[45] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[46] R. Mifflin. Semismooth and semieonvex functions in constrained optimization. *SlAM J. on Control and Optimization*, 15:957–972, 1977.

[47] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and variational problems. *CPAM*, XLII(5):577–685, 1988.

[48] Y. E. Nesterov. A method for solving the convex programming problem with convergence rate o(1/k2). *Dokl. Akad. Nauk SSSR,*, 269(11):543547, 1983.

[49] Yu. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, May 2005.

[50] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course (Applied Optimization)*. Springer Netherlands, 1 edition.

[51] Michael K. Ng, Liqun Qi, Yu-Fei Yang, and Yu-Mei Huang. On semismooth newton's methods for total variation minimization. *J. Math. Imaging Vis.*, 27(3):265–276, 2007.

[52] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2006.

[53] Stanley J. Osher and Selim Esedoglu. Decomposition of images by the anisotropic rudin-osher-fatemi model. *Comm. Pure Appl. Math*, 57:1609–1626, 2003.

[54] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An algorithm for minimizing the piecewise smooth mumford-shah functional. In *IEEE International Conference on Computer Vision (ICCV)*, Kyoto, Japan, 2009.

[55] L. D. Popov. A modification of the arrow-hurwicz method for search of saddle points. *Mathematical Notes*, 28(5):845–848, November 1980.

[56] R. T. Rockafellar. *Convex Analysis (Princeton Mathematical Series)*. Princeton University Press.

[57] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.

[58] Boyd S. and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[59] Christoph Schnörr. Unique reconstruction of piecewise-smooth images by minimizing strictly convex nonquadratic functionals. *Journal of Mathematical Imaging and Vision*, 4(2):189–198, May 1994.

[60] Petter Strandmark, Fredrik Kahl, and Niels Chr. Overgaard. Optimizing parametric total variation models. In *International Conference on Computer Vision*, 2009.

[61] A. N. Tikhonov and V. IA. Arsenin. *Solutions of ill-posed problems / Andrey N. Tikhonov and Vasiliy Y. Arsenin ; translation editor, Fritz John*. Winston ; distributed solely by Halsted Press, Washington : New York :, 1977.

[62] Dpartement Tsi, Jrme Darbon, Jrme Darbon, Marc Sigelle, and Marc Sigelle. A fast and exact algorithm for total variation minimization. In *In Pattern Recognition and Image Analysis: 2nd Iberian Conf., LCNS*, pages 351–359. Springer, 2005.

[63] C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM J. Sci. Comput*, 17:227–238.

[64] Mingliang Wang, Hector Klie, Manish Parashar, and Hari Sudan. Solving sparse linear systems on NVIDIA tesla GPUs. In *Computational Science ICCS 2009*, volume 5544 of *Lecture Notes in Computer Science*, pages 864–873. Springer Berlin / Heidelberg, 2009.

[65] Nesterov Yu. and Nemirovskii A. *Interior-Point Polynomial Algorithms in Convex Programming*. Philadelphia, PA, 1994.

[66] M. Zhu and T. Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. UCLA CAM Report 08-34, 2008.

[67] Mingqiang Zhu. *Fast numerical algorithms for total variation based image restoration*. PhD thesis, Los Angeles, CA, USA, 2008. Adviser-Chan, Tony F.

Mohamed Souiai
Oppelnder Str. 63
53119 Bonn

Hiermit erkläre ich, Mohamed Souiai, diese Diplomarbeit selbstständig durchgeführt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht zu haben.

Mohamed Souiai