Universität Regensburg
Fakultät für Mathematik

VARIATIONAL METHODS IN IMAGE PROCESSING USING A FAST PRIMAL-DUAL ALGORITHM

MASTERARBEIT

zur Erlangung des akademischen Grades
Master of Science
im Studiengang Computational Science

Eingereicht bei Prof. Dr. Harald Garcke
am 28.02.2016

Vorgelegt von:
Michael Bauer
Banater Str. 1
84061 Ergolsbach
Geboren am 23.08.1987 in Gräfelfing

Abstract

We present a comparison of different convex total variation based image processing algorithms and the non-convex Mumford-Shah Functional. We will revisit and make use of the fast primal-dual algorithm of [**?**]. We consider all steps, projections, operators and computations and will give a range of proofs. We also present a various range of applications and compare the presented methods to each other. Providing run-times, used memory and results will lead us to a conclusion which problem fits best to several situations.

Acknowledgements

Contents

1 Introduction

In the second decade of the twenty-first century autonomous driving seems to be the big thing for the Car, Computer and Software industry. The expectation is nothing less than having fewer, or even no, accidents. With these cars, industry is trying to change the world to a better. As this may comfort many people, it poses a lot of problems which need to be solved, for instance a stable hardware or even more important viable software. One field of research - of many others - is called Machine Learning, where the computer is trained to make the right decisions in the right situations. The car should be able to accommodate to all possible circumstances which could appear during a drive. To make it even more complicated, the car would need to take a decision in real-time. As one can imagine, it is extremely difficult to develop fast, stable and tractable methods.
Learning from a certain situation, the car and the computer respectively needs data from the environment. This can be the shape of another car, traffic lights, differences in the lighting conditions or the distance to other objects. What all these information have in common, they can be collected via images. Small cameras are tracking the traffic and surrounding and providing data. Using the images, the car's computer can be trained when to stop or where to turn right.
But it's not only the autonomously driving cars which make use of images to learn. Doctors use X-ray view or MRI scans in patient treatment, semiconductor companies use pictures of wafers seeking for damages on it, e.g. scratches or dark spots and unfortunately images are also used in modern warfare. For this reason image processing has become more important during the last decades. Researchers dealt with many problems like edge detection, deblurring, denoising, inpainting or image segmentation. In the field of edge detection the probably most famous researcher is John F. Canny. He provided the first tractable and stable edge detection algorithm, presented in the year 1986 (reference). One step in Canny's edge detector is to denoise and smooth the image. Therefore a Gaussian-Filter is used. The idea behind this filter is to convolute an image with a Gaussian kernel, or also called Gaussian curve. This filter provides reasonable results, but no more no less.
Only three years after Canny published his work, two researchers, namely David Mumford and Jayant Shah, published a paper whose impact continous to this day - to date it was cited over 4.500 times (scholar). They proposed to minimize the energy of the so called Mumford-Shah Functional in order to approximate an image optimally. Solving this optimization problem leads to applications like image denoising, inpainting and even segmentation could be handled. Unfortunately, this functional is by definition non-convex. For this reason finding the minimal energy of it is a NP-hard problem. This fact makes it so difficult to deal with but also so interesting.
One approach to compute the minimizer is by convex relaxation. It makes the non-

convex problem convex and one can apply a fast and tractable algorithm. The idea of convex relaxation goes back to Bouchite, Alberti, Dal Maso and was then used and further developed by Thomas Pock et.al. to make use of a fast primal-dual algorithm. It leads us to almost exact solutions and is highly parallelizable. Possibly inspired by the Mumford-Shah Functional another idea for image processing took place in 1992. Rudin, Osher and Fatemi focused on total variation based imaging. The total variation is a concept developed in the 19th century. Today there is a large community which associates total variation almost all with image processing. What Rudin et. al. proposed was also a minimization problem, namely the $TVL2$-Model (or ROF-Model). As the name suggests it is based on the total variation and the $L2$-Norm. The formulation of it, as we will see, looks quite similar to the Mumford-Shah Functional, but has one term left and is by definition convex. This makes it much easier to handle all computations but the output images are not even as good as they are applying Mumford-Shah. Applications which arise from the ROF-Model are rare. Removing noise from a given input image is the most common use. But as one later sees: Solutions are better than using a Gaussian-Filter.
By replacing the $L2$-Norm with the $L1$-Norm in the ROF-Model one derives the $TVL1$-Model, hence the name. A convex model and minimization its energy again is quite easy to compute. In this work we will see, that it is possibly the best model of the presented. It can deal with - so called - salt and pepper noise, handles inpainting, output images are sharp and close to the original image and most important: it runs in parallel on a GPU in real-time.
This work will present all of these methods, clarify which properties they have and make all computations clear. We also want to present the primal-dual algorithm which can be used to solve all models. We provide a large range of applications and the underlying run-times. At the end of this work we want to compare the models and conclude which one is the exactest, fastest and most tractable one.

2 Basic Concepts

In this first chapter we want to give an introduction to the most important concepts we meet in this thesis. We start by defining images in a mathematical manner, then introduce the total variation and basic concepts of convex analysis.

2.1 Images in Mathematics

Images can be viewed as mathematical objects. We can distinguish between discrete and continuous images. Let us first introduce what images are in the mathematical sense and then give some examples.

Definition 2.1 (Image) *Let $\Omega \subset \mathbb{R}^n$ be an image domain and C be a color space. A n-dimensional image u is a mapping $u : \Omega \longrightarrow C$, where each point in Ω corresponds to a color value in C.*

Remark 2.2 (Continuous and discrete images) *Yet, we did not tell anything about the difference between continuous and discrete images. We distinguish these two cases by the image domain Ω.*

1. *Let $\Omega = \{1, ..., N\}^n$, then u is a discrete image, since Ω is discrete.*
2. *Let $\Omega \subseteq \mathbb{R}^n$, then u is a continuous image. For instance, we could set $\Omega = [a, b]^n$ with $a, b \in \mathbb{R}$ and $a < b$.*

There are several image types like binary images, or colored images. The property to which class an image u belongs is determined by the color space C:

- *Let $C = \{0, 1\}$ then we call the image binary.*
- *In the case of grayscaled images we set $C = \{0, ..., 2^k - 1\}$, where the value k determines the bit depth. Usually we find in computers $k = 8$, i.e. grayscaled images take values in between 0 and 255, where $0 =$ black and $255 =$ white.*
- *n-dimensional color images have $C = \{0, ..., 2^k - 1\}^n$, or*
- *as a last example we want to see that an image u could also map to continuous color spaces, e.g. $C = [a, b]^n$ or $C = \mathbb{R}^n$. Most common are RGB (red-green-blue) images with $n = 3$, $a = 0$ and $b = 1$.*

In this thesis we meet two cases for the domain Ω. One where Ω is two-dimensional, and the other where Ω is a three-dimensional space. In the first case, we call a point (i, j) in Ω pixel, in the second case a point (i, j, k) is called voxel.

Computationally, it is a convenient method to store a two-dimensional image $u \in \mathbb{R}^{N \times M}$ not as a grid (like a matrix), but as a vector $u \in \mathbb{R}^{N \cdot M}$. To derive such a vector representation one needs to linearize the arguments of the function u. In other words we transform (i, j) to $(j + i \cdot N)$. The particular pixels are stored row wise, starting at the top left corner of the image and stopping at the bottom right corner. The access of one element becomes

$$u(i, j) \rightsquigarrow u(j + i \cdot N).$$

There are several programming languages where the indices of a vector start with 0. An example would be C++ and CUDA, which we used for our implementations. Then clearly, we have $u(0, 0) = u(0+0 \cdot N) = u(0)$ and $u(N-1, M-1) = u(M-1+(N-1) \cdot N)$. Even though we will treat u as a vector we still denote the discrete pixel positions with the notation

$$u(i, j) = u_{i,j} \quad \forall i = 1, ..., N, j = 1, ..., M.$$

Further, note that the notation $\langle \cdot, \cdot \rangle$ has two different meanings in this thesis. In the sense of infinite spaces, e.g. $u \in L^1(\Omega)$, the inner product of functions is defined by

$$\langle u, v \rangle = \int_\Omega u(x)v(x)dx.$$

Whereas, in a finite space, e.g. the euclidean space, this expression stands for the standard inner product (see also **??**).

2.2 Convex Optimization and Convex Analysis

In this section we cover a few topics which are related to convex analysis. Since we are considering (convex) optimization problems in this work we first want to define a convex program where we mainly follow [**?**], whereas convex sets and convex functions are the basis of convex optimization.

An optimization problem is of the form

$$\begin{array}{lll} \min\limits_{u \in \mathbb{R}^n} & F(u) & \\ \text{subject to} & G_i(u) \leq 0, & i = 1, ..., m \\ & H_j(u) = 0 & j = 1, ..., p, \end{array} \tag{2.1}$$

where

- $u \in \mathbb{R}^n$ is called the *optimization variable*,
- $F : \mathbb{R}^n \longrightarrow \mathbb{R}$ *objective function* or *cost function*,
- $G_i(u) \leq 0$ *inequality constraints* for all $i = 1, ..., m$,
- $H_j(u) = 0$ *equality constraints* for all $j = 1, ..., p$,
- $G_i : \mathbb{R}^n \longrightarrow \mathbb{R}$ for all $i = 1, ..., m$ the *inequality constraint functions* and

- $H_j(u) = 0$ for all $j = 1, ..., p$ the *equality constraint functions.*

If we have no constraints, meaning $m = p = 0$, we say the problem **??** is *unconstrained.* It is called convex if all, the objectiv, equality and inequality functions, are convex.
Further, we define the domain of the optimization problem **??** by

$$\mathcal{D} = \operatorname{dom}(F) \cap \bigcap_{i=1}^{m} \operatorname{dom}(G_i) \cap \bigcap_{j=1}^{p} \operatorname{dom}(H_j). \tag{2.2}$$

This set is the set of points for which the objective and all constraint functions are defined. We call a point $u \in \mathcal{D}$ *feasible* if it satisfies all constraints. The optimization problem **??** is said to be feasible if there exists at least one feasible point, and *infeasible* otherwise. We call the set of all feasible points the *feasible set* or alternatively the *constraint set.*
We define the optimal value u^* of the problem **??** by

$$u^* = \inf \big\{ F(u) : G_i(u) \leq 0, i = 1, ..., m, H_j(u) = 0, j = 1, ..., p \big\}, \tag{2.3}$$

where we allow u^* to take on the extended values $\pm\infty$. We set $u^* = \infty$ if the problem is infeasible, since $\inf(\emptyset) = \infty$. If there are feasible points u_k with $F(u_k) \longrightarrow -\infty$ as $k \longrightarrow \infty$, then $u^* = -\infty$, and we say that problem **??** is *unbounded below.*

Since, convex optimization is based on convex functions we need to define these class of functions and its properties. We set $X \subseteq \mathbb{R}^n$ and its dual space $X^* = Y$. According to [**?**] and [**?**] we have the following basic definitions, propositions and theorems.

Definition 2.3 (Convex Set) *A subset $C \subseteq X$ is said to be convex if and only if for any $u_1, u_2 \in C$, the line segment $[u_1, u_2] \subseteq C$, that is, for any $t \in [0, 1]$,*

$$u_1 t + u_2(1 - t) \in C. \tag{2.4}$$

We call C strictly convex, if it is closed and

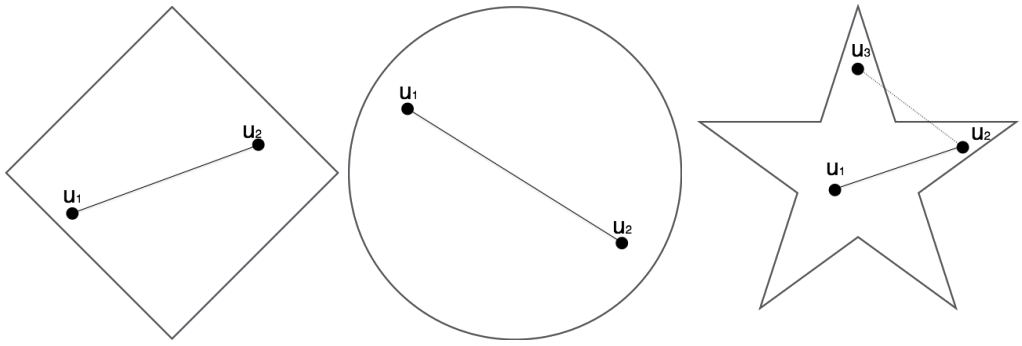
$$(1 - t)u_1 + u_2 t \in \operatorname{int} C, \;\; \forall u_1, u_2 \in C, \;\; u_1 \neq u_2, \;\; t \in [0, 1], \tag{2.5}$$

where int *stands for interior.*

This definition ensures that if C is convex, we can always find two arbitrary points in C such that the line segment $[u_1, u_2]$ with end points u_1, u_2 lies fully in C (see Figure **??**).

Definition 2.4 (Inidcator Function, Support Function) *For any subset $C \subseteq X$ of a vector space, the indicator function $\delta_C : X \longrightarrow \mathbb{R}_\infty$ is defined as*

$$\delta_C(u) = \begin{cases} 0 & \text{if } u \in C, \\ \infty & \text{if } u \notin C. \end{cases} \tag{2.6}$$



(a) l_1 unit sphere (b) l_2 unit sphere (c) Star: a non-convex set.

Figure 2.1: We see three different sets, where (a) and (b) are convex, but (c) is not. (a) refers to the unit sphere of the l_1 norm where (b) refers to the unit sphere of the l_2 norm. In (c) we see that the line segment of $[u_2, u_3]$ lies not in the set itself.

The support function of the set C is defined as

$$S_C(u) = \sup_{p \in C} \langle p, u \rangle, \tag{2.7}$$

where we allow $S_C(u)$ to be $+\infty$.

Definition 2.5 (Convex Function, Proper, Lower-Semicontinuouity) *A function $F : X \longrightarrow \mathbb{R}_\infty$ is said to be*

- *convex if for all $u_1, u_2 \in C$ and any $t \in [0, 1]$ the inequality*

$$F(u_1 t + u_2(1-t)) \leq F(u_1)t + F(u_2)(1-t) \tag{2.8}$$

  *is satisfied. F is called strictly convex, if the inequality **??** holds strictly whenever u_1, u_2 are distinct points and $t \in (0, 1)$.*

- *proper if F is not identically $-\infty$ or $+\infty$.*

- *lower-semicontinuous (l.s.c) if for any $u \in X$ and a sequence (u_n) converging to u,*

$$F(u) \leq \liminf_{n \to \infty} F(u_n). \tag{2.9}$$

We define $\Gamma_0(X)$ as the set of all convex, proper, l.s.c. functions on X.

A common example for a strictly convex function would be a quadratic function (see Example **??**). In Figure **??** (a) this is also illustrated, together with the plot in (b) of the function

$$F(u) = \begin{cases} u^2 & x \in [-1, 1] \\ -\frac{1}{2}u(u-4) & x \in (1, 3]. \end{cases} \tag{2.10}$$

being lower-semicontinuous.

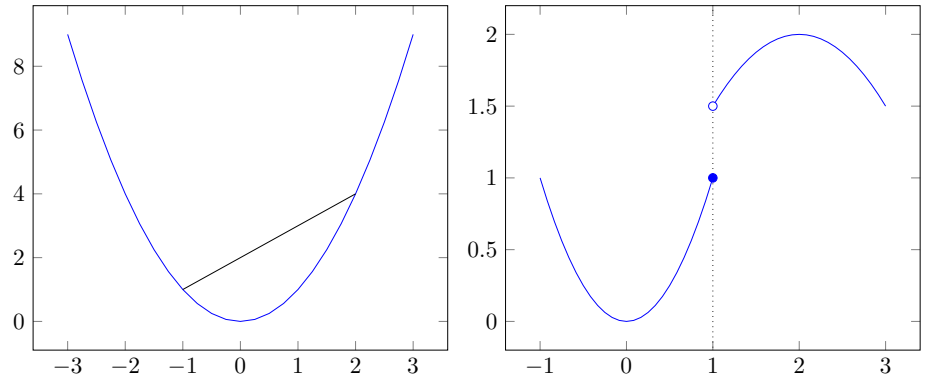
Proposition 2.6 *Let $F, G : X \longrightarrow \mathbb{R}_\infty$ be two convex functions. Then $F + G$ is also convex.*

Proof This can easily be verified. Define $H(u) = F(u) + G(u)$ with F, G convex and choose $u_1, u_2 \in X$ and $t \in [0, 1]$, then we obtain

$$\begin{aligned} H(u_1 t + u_2(1-t)) &= F(u_1 t + u_2(1-t)) + G(u_1 t + u_2(1-t)) \\ &\leq F(u_1)t + F(u_2)(1-t) + G(u_1)t + G(u_2)(1-t) \\ &= \underbrace{F(u_1)t + G(u_1)t}_{=H(u_1)t} + \underbrace{F(u_2)(1-t) + G(u_2)(1-t)}_{H(u_2)(1-t)} \\ &= H(u_1)t + H(u_2)(1-t), \end{aligned}$$

which shows that H is convex. ■

Remark 2.7 (Concave Function) *Let F be as in Definition* **??**. *If $-F$ is (strictly) convex, then we say that F is (strictly) concave. If F is both convex and concave we say that F is affine, i.e. equality holds in Equation* **??**.



(a) A quadratic function. (b) A l.s.c. function.

Figure 2.2: (a) shows one line segment above a convex (quadratic) function which lies completely in epi(F). (b) is the l.s.c. function of Equation **??**. The left part of the function accesses the value at $(1, 1)$, but the right one does not.

Example 2.8 *1. Let $A \in \mathbb{R}^{m \times n}$ be a real matrix and $F : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ a linear function with $F(u) = Au$. Then F is convex and concave, hence linear functions are affine.*

Proof *Choose $u_1, u_2 \in \mathbb{R}, t \in [0, 1]$. We get*

$$F(u_1 t + u_2(1-t)) = F(u_1 t) + F(u_2(1-t)) = F(u_1)t + F(u_2)(1-t)$$

by definition of linearity. ■

2. *Let $A \in \mathbb{R}^{n\times n}$ be a real, symmetric, positiv definite matrix and $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ a linear function with $F(u) = \frac{1}{2}u^T Au + b^T u + c$. Then F is strictly convex. To show this, we make use of a fundamental result of the calculus class. Since, if $\mathcal{H}\big(F(u)\big) > 0$ then the function F is strictly convex. Here $\mathcal{H}$ stands for the Hessian of the function F. We have*

$$\mathcal{H}\big(F(u)\big) = A > 0$$

by definition of A being symmetric, positiv definite.

3. *This example will later be useful: Each norm $||\cdot|| : X \longrightarrow \mathbb{R}$ on a normed vector space X is convex (see also Figure **??** for the l_1 and l_2 norms).*

Proof *Choose $u, v \in X, t \in [0,1]$, then*

$$||ut + v(1-t)||_X \overbrace{\leq}^{(*)} ||ut||_X + ||v(1-t)||_X \overbrace{=}^{(**)} ||u||_X t + ||v||_X (1-t).$$

In $(*)$ we used the triangle inequality and in $(**)$ absolute homogeneity with the fact that $t \in [0,1]$. ■

Sometimes in literature you find another definition of convex functions. Therefore let us first introduce the epigraph and the domain of a function.

Definition 2.9 (Domain, Epigraph) *For any function $F : X \longrightarrow \mathbb{R}_\infty$, we define the domain*

$$\mathrm{dom}(F) = \{u \in X : F(u) < +\infty\}, \tag{2.11}$$

and the epigraph

$$\mathrm{epi}(F) = \{(u,t) \in X \times \mathbb{R} : t \geq F(u)\} \in \mathbb{R}^{n+1}. \tag{2.12}$$

Then we have the following definition for a convex function.

Definition 2.10 (Convex Function) *A function $F : X \longrightarrow \mathbb{R}_\infty$ is convex if* $\mathrm{dom}(F)$ *and* $\mathrm{epi}(F)$ *are convex sets.*

Definition 2.11 (Closed Function) *Let $F : X \longrightarrow \mathbb{R}_\infty$ be a function. Then F is closed if* $\mathrm{epi}(F)$ *is a closed set.*

Now, that we are set up with convexity, we want to introduce one of the most important concepts for this thesis.

Definition 2.12 (Legendre-Fenchel conjugate) *Let $F : X \longrightarrow \mathbb{R}_\infty$ be a convex function. We define the Legendre-Fenchel conjugate F^* of F for any $p \in X^*$ by*

$$F^*(p) = \sup_{u\in X} \big(\langle p, u\rangle - F(u)\big). \tag{2.13}$$

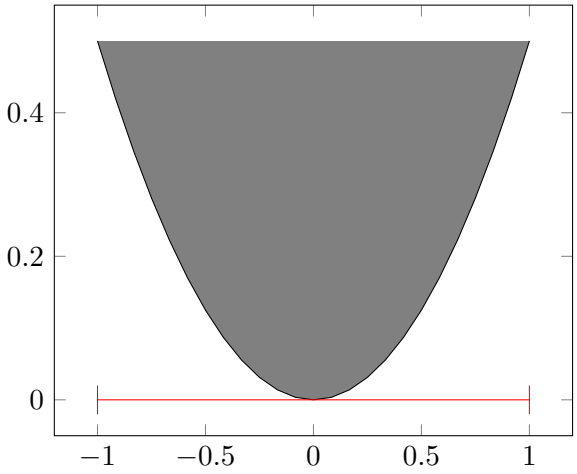


Figure 2.3: The domain (red) denoted as $\mathrm{dom}(F) = [-1, 1]$ and epigraph $\mathrm{epi}(F) = \{(u, t) \in [-1, 1] \times \mathbb{R} : t \geq \frac{1}{2}u^2\}$ (gray) of a function $F : [-1, 1] \longrightarrow \mathbb{R}$ with $F(u) = \frac{1}{2}u^2$.

Remark 2.13 *Without a proof we state that F^* - as the supremum of linear, continuous functions - is convex and lower-semicontinous, even F is not. In addition F^* is proper if F is convex and proper.*

Theorem 2.14 *Let $F \in \Gamma_0(X)$, then $F^{**} = F$.*

The last theorem assures that we can rewrite Equation **??** to derive

$$F(u) = \big(F^*(p)\big)^*(u) = \sup_{p \in Y} \big(\langle u, p\rangle - F^*(p)\big).$$

Example 2.15 *Let us view some examples on the Legendre-Fenchel conjugate.*

1. *The Legendre-Fenchel conjugate of the indicator function of a set $C \subseteq X$ is given by*
$$\delta_C^*(p) = \sup_{u \in C} \langle p, u\rangle - \delta_C(u) = \sup_{u \in C} \langle p, u\rangle,$$
which is the support function.

2. *Let F be an arbitrary function on X and $\alpha > 0$. Then the convex conjugate of $\alpha F(u)$ is given by*
$$F^*(p) = \alpha F^*(\frac{p}{\alpha}).$$
To show this we compute
$$F^*(p) = \sup_{u \in X} \langle p, u\rangle - \alpha F(u) = \alpha\big(\underbrace{\sup_{u \in X} \langle \frac{p}{\alpha}, u\rangle - F(u)}_{=F^*(\frac{p}{\alpha})}\big) = \alpha F^*(\frac{p}{\alpha}).$$

3. *Now, let $||\cdot||$ be a norm on X, with dual norm $||\cdot||_*$ on Y. We show that*

$$F^*(p) = \begin{cases} 0 & \text{if } ||p||_* \leq 1, \\ \infty & \text{else,} \end{cases}$$

i.e. the conjugate of a norm is the indicator function of the unit ball of its dual norm. We have

$$F^*(p) = \sup_{u \in X} (\langle p, u \rangle - ||u||).$$

First let $||p||_* > 1$. By definition of the dual norm ($||p||_* = \sup_{||x|| \leq 1} |\langle p, x \rangle|$) there is a $x \in \mathbb{R}^n$ and we observe $\langle p, x \rangle > 1$.

We take $u = tx$ and let $t \longrightarrow \infty$, then we get

$$\langle p, u \rangle - ||u|| = t(\langle p, x \rangle - ||x||) \longrightarrow \infty,$$

which shows that $F^*(p) = \infty$. On the other hand if $||p||_* \leq 1$ Cauchy-Schwarz-inequality assures that

$$\langle p, u \rangle \leq ||u||\, ||p||_*$$

for all $u \in X$. But this implies

$$\langle p, u \rangle - ||u|| \leq ||u||\, ||p||_* - ||u|| = ||u||\, (||p||_* - 1) \leq 0.$$

Because $||p||_* \leq 1$ holds, to get the supremum we need to choose $u = 0$. This shows that $F^* = \sup_{u \in X} \langle p, 0 \rangle - ||0|| = 0$.

4. *As a last example we show that for a function $F(u) = \frac{1}{2}||u||^2$ its conjugate is $F^*(p) = \frac{1}{2}||p||_*^2$. With Cauchy-Schwarz-inequality we have $\langle p, u \rangle \leq ||p||_*\, ||u||$. We observe*

$$\langle p, u \rangle - \frac{||u||^2}{2} \leq ||p||_*\, ||u|| - \frac{||u||^2}{2},$$

where the righthand side is a quadratic function of $||u||$. Define $H(x) := -\frac{x^2}{2} + y\, x$, then the maximal value of H is $x = y$, since $H^{'}(x) = -x + y$ and $H^{'}(x) = 0$ implies $x = y$. Plugging $||u||$ into H we see, that the maximum is attained at $||p||_*$. It follows

$$\langle p, u \rangle - \frac{||u||^2}{2} \leq ||p||_* ||p||_* - \frac{||p||_*}{2} = \frac{||p||_*^2}{2}.$$

This shows that $F^*(p) \leq \frac{||p||_*^2}{2}$. Let us now proof the inequality in the other direction. We assume that we find an arbitrary u which satisfies $\langle u, p \rangle = ||u||\, ||p||_*$. Further, we assume that $||u|| = ||p||_*$ where u is scaled so that the last equation holds. Then we have

$$\langle p, u \rangle - \frac{||u||^2}{2} = ||u||\, ||p||_* - \frac{||u||^2}{2} = \frac{||p||_*^2}{2},$$

but for this particular u it holds that

$$\langle p, u\rangle - \frac{||u||^2}{2} \leq \sup_{u\in X}\langle p, u\rangle - \frac{||u||^2}{2} = F^*(p).$$

From these two inequalities it follows that $F^*(p) = \frac{||p||_*^2}{2}$

Another important property of functions is differentiability. Unfortunatelly, in some cases a function F is not differentiable everywhere. For this, we want to define the so called subdifferential and therefore the subgradient.

Definition 2.16 (Subgradient, Subdifferential) *Let X be an open, convex set and $F : X \longrightarrow \mathbb{R}_\infty$ a (convex) function. A vector y is called subgradient of F in $u_0 \in X$, if*

$$F(u) \geq F(u_0) + \langle y, u - u_0\rangle \ \ \forall u \in X. \tag{2.14}$$

The set

$$\partial F(u_0) = \{y \in X : \ F(u) \geq F(u_0) + \langle y, u - u_0\rangle \ \ \forall u \in \mathrm{dom}(F)\} \tag{2.15}$$

is called subdifferential of F in $u_0 \in X$ and $\mathrm{dom}(\partial F) = \{u : \partial F(u) \neq \emptyset\} \subset F$.

If F and G are differentiable we have $\partial(F+G) = \partial F + \partial G$. For the subdifferential this holds under some conditions. We want to give the corresponding Proposition without giving a proof. We just state that in our computations this equality can always be used.

Proposition 2.17 *Let F, G be convex and assume* $\mathrm{int}(\mathrm{dom}G) \cap \mathrm{dom}F \neq \emptyset$*: then*

$$\partial(F + G) = \partial F + \partial G.$$

To illustrate the subgradient and subdifferential, respectively, we want to give some examples.

Example 2.18 *1. Take the absolute value function $F(u) = |u|$ in $\mathbb{R}$ which is defined by*

$$F(u) = \begin{cases} u & \text{if } u \geq 0, \\ -u & \text{else.} \end{cases}$$

Since $F(u)$ is not differentiable in 0, but on $\mathbb{R}/\{0\}$ we compute the subgradient y as

- $\partial F(u) = 1$ *if* $u > 0$,
- $\partial F(u) = -1$ *if* $u < 0$,
- *and finally*

$$F(0) + y(u - 0) \leq F(u) \Longleftrightarrow \langle y, u\rangle \leq |u|.$$

If $u \geq 0$ this is equivalent to $y \geq 1$. If $u < 0$ we get

$$\langle y, u\rangle \leq |u| \Longleftrightarrow \langle y, u\rangle \leq -u \Longleftrightarrow y \geq -1.$$

Finally, we have

$$y = \begin{cases} 1 & \text{if } u > 0, \\ -1 & \text{if } u < 0, \\ [-1,1] & \text{if } u = 0. \end{cases}$$

2. *Let $F(u) = ||u||_1$. As a non-differentiable convex function we seek for the subdifferential. For that we can express the l_1-norm as*

$$||u||_1 = |u_1| + ... + |u_n| = \max\big\{\langle y, u\rangle : p_i \in \{-1, 1\}\big\},$$

for all $u \in X$. Obviously, one gets $||y||_\infty \leq 1$. If we find a y such that $\langle y, u\rangle = ||u||_1$ then immediately inequality **??** *would be satisfied, because*

$$F(u) \geq F(u_0) + \langle y, u\rangle \iff ||u||_1 \geq 0 + \langle y, u\rangle = ||u||_1.$$

On the other hand, if we choose $p_i = -1$ if $u_i < 0$ and $p_i = 1$ if $u_i > 0$ (this was also what we got by calculating the subgradient of the absolute value function) we only have the case left where $u_i = 0$. In this case, we can choose both $p_i = -1$ or $p_i = 1$ or equivalently looking at the non-differentiable point $u_0 = 0$ and if $||y||_\infty \leq 1$, we observe

$$F(u) \geq F(0) + \langle y, (u - 0)\rangle \iff ||u||_1 \geq \langle y, u\rangle.$$

This means we have

$$y = \begin{cases} 1 & \text{if } u > 0, \\ -1 & \text{if } u < 0, \\ -1 \text{ or } 1 & \text{if } u = 0. \end{cases}$$

and

$$\partial F(u) = \big\{y : ||y||_\infty \leq 1,\ \langle y, u\rangle = ||x||_1\big\}. \tag{2.16}$$

The following Propositions are elementary for computations later. We provide them without giving a proof and refer to [**?**] and [**?**].

Proposition 2.19 *Let $F \in \Gamma_0(X)$. Then*

- *the set of minimizers $\arg\min_{u \in X} F(u)$ is convex (possibly empty).*
- *if $\hat{u}$ is a local minimum of F, then $\hat{u}$ is in fact a global minimum, i.e.*

$$\hat{u} \in \arg\min_{u \in X} F(u).$$

Proposition 2.20 *For any F convex, $p \in \partial F(u)$ if and only if*

$$\langle p, u\rangle - F(u) = F^*(p).$$

Moreover if $F \in \Gamma_0$, so that $F^{**} = F$, then this is equivalent to $u \in \partial F^*(p)$.

Proposition 2.21 *Let F be convex, then $\hat{u} \in \arg\min_{u \in X} F(u)$ if and only if $0 \in \partial F(\hat{u})$.*

The three propositions from above are fundamental for our work. If we find a (local) minimizer of a convex optimization problem, we already know that this minimizer is the global optimum. Further, we see in Propositions **??** how the variables u and p can be interchanged by using the subdifferential. The last statement assures, that if we find a global minimum of our convex optimization problem, then we have that 0 always is an element of the subdifferential.
The last theorem, we will provide, can be found in [**?**] and is used extensively in the following chapters. One can find different editions of the theorem, see for instance [**?**]. We provide it without a proof, for which we also refer to [**?**]. But, first we need:

Definition 2.22 (Projection Operator, Proximity Operator, Moreau Envelop) *For any non-empty closed set C the projection operator for an arbitrary point $z \notin C$ on the set C is defined by*

$$\Pi_C(z) = \arg\min_{u \in C} \frac{1}{2}||z - u||_2^2, \tag{2.17}$$

meaning the orthogonal projection onto the convex set C. We define the Proximity Operator in a similar fashion due to

$$\text{prox}_F^\alpha(z) = \arg\min_{u \in X} \frac{1}{2}||u - z||_2^2 + \alpha\, F(u), \tag{2.18}$$

and the Moreau envelop as

$$M_F^\alpha(z) = \min_{u \in X} \frac{1}{2}||u - z||_2^2 + \alpha\, F(u), \tag{2.19}$$

for any $\alpha > 0$.

The Proximity operator is a strict generalization of the Projection operator, while the Moreau envelop is a generalization of the squared distance measure function. Later, we will work out the connection between the projection and the proximity operator in detail. Let us shortly view two useful examples for the projection operator.

Example 2.23 *The l_p-Projection of a point $p \in \mathbb{R}^n$ onto the l_p-unit sphere with $p \notin C = \{u : ||u||_p \leq 1\}$ is given by the following minimization problem:*

$$\begin{array}{ll} \min\limits_{u \in C} & \frac{1}{2}||u - p||_2^2 \\ \text{subject to} & ||u||_p \leq 1. \end{array} \tag{2.20}$$

We are looking at the l_2-Projection (euclidean projection) and the l_∞-Projection, where both convex optimization problems have unique solutions, for more information see for instance

1. *The solution for the euclidean projection is given by*

$$u^* = \begin{cases} p, & \text{if } ||p||_2 \leq 1 \\ \frac{p}{||p||_2}, & \text{otherwise.} \end{cases} \tag{2.21}$$

Or equivalently one gets $\Pi_{l_2 \leq 1}(u) = \frac{u}{\max(1,|u|)}$.

2. *For the* l_∞*-Projection we observe*

$$u_k^* = \begin{cases} p_k, & \text{if } |p_k| \leq 1 \\ 1, & \text{otherwise,} \end{cases} \tag{2.22}$$

for $k = 1, ..., n$ *or* $\Pi_{l_\infty \leq 1}(u) = \min(1, \max(-1, u))$.

Theorem 2.24 (Moreau's Theorem) *Let* $F \in \Gamma_0(X)$ *and* $F^*(p) = \sup_{u \in X} \langle u, p \rangle - F(u)$ *be its Legendre-Fenchel conjugate. Then*

$$M_F^\alpha(z) + M_{F^*}^\alpha(z) = \frac{1}{2}||z||^2, \tag{2.23}$$

i.e. for each $z \in \mathbb{R}^n$ *one has*

$$\min_{u \in X} \frac{1}{2}||u - z||^2 + \alpha\, F(u) + \min_{p \in X^*} \frac{1}{2}||p - z||^2 + \alpha\, F^*(p) = \frac{1}{2}||z||^2, \tag{2.24}$$

where both minima are uniquely attained. The unique vectors u *and* p *for which the respective minima are attained for a given* z *are the unique vectors* u *and* p *such that*

$$z = u + p, \quad p \in \partial F(u), \tag{2.25}$$

and they are given by

$$u = \text{prox}_F^\alpha(z), \quad p = \text{prox}_{F^*}^\alpha(z). \tag{2.26}$$

This theorem was first proven in 1965 by Jean J. Moreau in (Moreau paper) and is based on the theorem of the closed complement. First Moreau generalized the closed complement theorem in 1962 and three years later stated the celebrated Theorem **??**.

Remark 2.25

- *Note that Equation* **??** *is equivalent to* $u \in \partial F^*(p)$ *or* $F(u) + F^*(p) = \langle u, p \rangle$*, by Proposition* **??**.
- *Another way to denote the proximity operator is given by*

$$\text{prox}_F^\alpha = (\text{Id} + \alpha \partial F)^{-1}. \tag{2.27}$$

This notation can be derived by Proposition **??**. *Here, zero is an element of the subdifferential of Equation* **??**, *namely*

$$0 \in \partial\big(\alpha F(u) + \frac{1}{2}||u - z||_2^2\big) = \alpha\, \partial F(u) + u - z.$$

But this can be rewritten as

$$z \in \alpha\,\partial F(u) + u \Longleftrightarrow u = (\mathrm{Id} + \alpha\,\partial F)^{-1}(z).$$

For that, the solution u is well-defined and unique. Furthermore, using Equation **??** *one gets*

$$u = (\mathrm{Id} + \alpha\,\partial F)^{-1}(u) + \alpha\left(\mathrm{Id} + \frac{1}{\alpha}\partial F^*\right)^{-1}\left(\frac{u}{\alpha}\right). \tag{2.28}$$

*We refer to [**?**] and [**?**] for more information.*

Example 2.26 *To this end we want to briefly discuss one example for the proximity operator. If we set $F(u) = \delta_C(u)$ then we observe*

$$\mathrm{prox}_F^\alpha(\tilde{u}) = \min_{u\in C} \frac{||u-\tilde{u}||_2^2}{2} + \alpha\,\delta_C(u) = \min_{u\in C} \frac{||u-\tilde{u}||_2^2}{2} = \Pi_C(\tilde{u}).$$

Here, one can see the connection between the two operators. Proximation of the indicator function is the same as doing a euclidean projection onto the corresponding set C.

2.3 Total Variation

The definitions of the total variation, varition of functions and functions of bounded variation can be found in [**?**], together with the first example in this section.

Definition 2.27 (Total Variation) *Let Ω be an open subset of $\mathbb{R}^n$. For a function $u \in L^1(\Omega)$, the* total variation *of u in Ω is defined as*

$$\mathrm{TV}(u) = \sup\left\{ -\int_\Omega u \,\mathrm{div}\,\varphi\, dx : \varphi \in C_0^1(\Omega, \mathbb{R}^n), |\varphi(x)| \le 1, \forall x \in \Omega \right\}. \tag{2.29}$$

Example 2.28 *Let $u \in W_1^1(\Omega)$, then integration by parts and the fact that φ having compact support leads to*

$$\begin{aligned} -\langle u, \mathrm{div}\,\varphi\rangle &= -\int_\Omega u\,\mathrm{div}\,\varphi\, dx \\ &= -\Big(\underbrace{\int_{\partial\Omega} u\,\varphi\, d\mathcal{H}^{n-1}}_{=0} - \int_\Omega \nabla u\,\varphi\, dx\Big) = \int_\Omega \nabla u\,\varphi\, dx \\ &= \langle \nabla u, \varphi\rangle, \end{aligned} \tag{2.30}$$

for every $\varphi \in C_0^1(\Omega, \mathbb{R}^n)$ so that

$$\mathrm{TV}(u) = \int_\Omega |\nabla f| dx. \tag{2.31}$$

The idea to derive Equation **??** *is that we first evaluate the case where* $\mathrm{TV}(u) \leq \int_\Omega |\nabla f| dx$. *This can easily be seen by using Cauchy-Schwarz-inequality and*

$$\begin{aligned} \int_\Omega \nabla u \, \varphi \, dx &\leq \left| \int_\Omega \nabla u \, \varphi dx \right| dx \\ &\leq \int_\Omega |\nabla u| \underbrace{|\varphi|}_{\leq 1} dx \leq \int_\Omega |\nabla u| \, dx. \end{aligned} \tag{2.32}$$

For the inequality in the other direction the key idea would be to set $\varphi = \frac{\nabla u}{|\nabla u|}$. *Then clearly,* $\varphi = 1$. *Since* $\varphi \notin C_0^1$, *we would approximate* φ *in* L^1. *This space lies close in the space of smooth functions with compact support and for that satisfies the assumptions.*

Remark 2.29 *Note that* $W_1^1(\Omega)$ *denotes the* L^1*-Sobolev-Space of* Ω, *which means that* $u \in L^1(\Omega)$ *and* $Du \in L^1(\Omega)$. *Here* Du *is meant in the sense of distributional derivatives. From now on, we assume that all images* u *are in* $W_1^1(\Omega)$, *where* Ω *is the image domain, so that Equation* **??** *holds.*
Further, the total variation is convex and lower-semicontinous. See [?] for details.

To get a better intuition about the Total Variation, we also want to give the defintion of the variation of functions $u \in [a, b] \subset \mathbb{R}$ with $a, b \in \mathbb{R}$ and $a < b$.

Definition 2.30 (Variation of a function) *Let* $u : \mathbb{R} \longrightarrow \mathbb{R}$ *and* $a < b$ *be real numbers. Then define the variation of* u *on* $[a, b]$ *as*

$$V_a^b(u) = \sup \left\{ \sum_{i=1}^n |u(x_i) - u(x_{i-1})| : m \in \mathbb{N} \, \text{and} \, a = x_0 < x_1 < ... < x_m = b \right\}.$$

This definition is well known and suited for functions form $\mathbb{R}$ to $\mathbb{R}$. In the sense of measures and integrals this definition is ill-posed (referring to [**?**]). But to make variation a bit more vivid this definition is perfect.

Example 2.31 *1. Let* $a, b \in \mathbb{R}_{>0}$ *with* $a < b$. *If* $u : [a, b] \longrightarrow \mathbb{R}$ *is monotonically increasing, then for any* $a = x_0 < x_1 < ... < x_n = b$ *we observe*

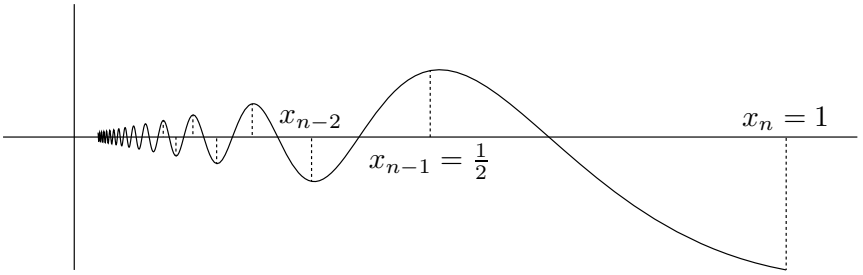
$$\begin{aligned} & \sum_{i=1}^n |u(x_i) - u(x_{i-1})| = \sum_{i=1}^n u(x_i) - u(x_{i-1}) \\ = \; & u(x_1) - u(x_0) + u(x_2) - u(x_1) + ... + u(x_{n-1}) - u(x_{n-2}) + u(x_n) - u(x_{n-1}) \\ = \; & u(x_n) - u(x_0) = u(b) - u(a). \end{aligned}$$

It follows that $V_a^b(u) = \sup\{u(b) - u(a)\} = u(b) - u(a)$.

2. Define the function $u : [0, 1] \longrightarrow \mathbb{R}$ *by*

$$u(x) = \begin{cases} 0, & \text{if } x = 0, \\ x \cos(\frac{\pi}{x}), & \text{if } x \neq 0. \end{cases} \tag{2.33}$$

This function is continuous and we have $V_a^b(u) = \infty$. To see this, consider for each $m \in \mathbb{N}$, the partition $P_m = \{0, \frac{1}{2m}, \frac{1}{2m-1}, ..., \frac{1}{3}, \frac{1}{2}, 1\}$. The values of u at the points of this partition are $u(P_m) = \{0, -\frac{1}{2m}, \frac{1}{2m-1}, ..., -\frac{1}{3}, \frac{1}{2}, -1\}$.



For this partition,

$$\begin{aligned}\sum_{i=1}^{n} |u(x_i) - u(x_{i-1})| &= |\frac{1}{2m} - 0| + |-\frac{1}{2m-1} - \frac{1}{2m}| + ... + |\frac{1}{2} + \frac{1}{3}| + |-1 - \frac{1}{2}| \\ &= \frac{1}{2m} + \frac{1}{2m-1} + \frac{1}{2m} + \frac{1}{2m-1} + ... + \frac{1}{2} + \frac{1}{3} + 1 + \frac{1}{2} \\ &= 2 * (\frac{1}{2m} + \frac{1}{2m-1} + ... + \frac{1}{2}) + 1.\end{aligned}$$

It is known, that the (harmonic) series $\sum_{k=2}^{\infty} \frac{1}{k}$ diverges. So given any m the partition P_m always ensures

$$V_a^b(u) = \sup\{2 * \sum_{k=2}^{\infty} \frac{1}{k} + 1\} = \infty.$$

Another concept which later is used in Chapter 3 are (special) functions of bounded variation. Even though we do not expliciely make use of this class of functions, we want to give the definition(s) for the sake of completeness. Further, (special) funcitons of bounded variation are strongly related to Total Variation as the following definition (see for instance [**?**]) ensures.

Definition 2.32 (Functions of bounded variation) *A function $u \in L^1(\Omega)$ is said to have bounded variation in Ω if $\mathrm{TV}(u) < \infty$. We define $BV(\Omega)$ as the space of all functions in $L^1(\Omega)$ with bounded variation. This space is equipped with the norm*

$$||u||_{BV} = ||u||_{L^1} + \int_\Omega |Du|, \tag{2.34}$$

and for that it is a Banach space.

As mentioned the Total Variation is the basis for this class of functions. Related to example **??** we find that $u \in BV(\mathbb{R})$ in the first example, since we had $V_a^b(u) = u(b) - u(a) < \infty$. Furthermore u in the second example is not a function of bounded variation, because $V_a^b(u) = \infty$.
Additionally, BV-functions are proper and therefore we have if $u \in BV(\Omega)$, then also $u \in \Gamma_0(\Omega)$, since the total varition is convex and l.s.c.

Definition 2.33 (Special functions of bounded variation [?]) *SBV denotes the special functions of bounded variation, i.e. functions u of bounded variation for which the derivative Du is the sum of an absolutely continuous part $\nabla u \cdot dx$ and a discontinuous singular part S_u.*

These classes of functions play indeed an important role for the Mumford-Shah Functional. But in this thesis the definition is only given for the sake of completeness.

3 A Primal-Dual Algorithm for (Convex) Saddle-Point Problems

3.1 The General Saddle-Point Problem

In this first section we will present the general class of problem we consider in this work. Therefore we define the the map $K : X \longrightarrow Y$ as a continuous linear operator with induced norm

$$||K|| = \max\big\{||Kx||_Y : x \in X \text{ with } ||x||_X \leq 1\big\}. \tag{3.1}$$

Furthermore, we set the map $K^* : Y \longrightarrow X$ as the adjoint operator of K.
Now, let K be a linear operator, $u \in X$, $p \in Y$ and define $G : X \longrightarrow \mathbb{R}_+$ and $F^* : Y \longrightarrow \mathbb{R}_+$ where $G, F^* \in \Gamma_0$ and F^* being the Legendre-Fenchel conjugate of a convex, l.s.c. function F. We are trying to find the saddle point (u, p) of the following problem:

$$\min_{u\in X}\max_{p\in Y}\langle Ku, p\rangle + G(u) - F^*(p). \tag{3.2}$$

We call this problem also the primal-dual-problem, where u is the primal and p the dual variable. We define the corresponding primal-problem to this formulation by

$$\min_{u\in X} F(Ku) + G(u), \tag{3.3}$$

and the dual-problem by

$$\max_{p\in Y} -(G^*(-K^*p) + F^*(p)). \tag{3.4}$$

These three different classes of problems are equivalent since the Legendre-Fenchel conjugate assures that

$$\begin{aligned}
\min_{u\in X} \underbrace{F(Ku)}_{=\sup_{p\in Y}\langle p,Ku\rangle - F^*(p)} + G(u) &= \min_{u\in X}\max_{p\in Y}\langle p, Ku\rangle - F^*(p) + G(u) \\
&= \max_{p\in Y}\min_{u\in X}\langle K^*p, u\rangle + G(x) - F(p) \\
&= \max_{p\in Y} \underbrace{\max_{u\in X} -\big(\langle K^*p, u\rangle + G(x)\big)}_{=-(G^*(-K^*)p)} - F(p) \\
&= \max_{p\in Y} -(G^*(-K^*p) + F^*(p))
\end{aligned} \tag{3.5}$$

Note that we can swap min and max and exchange sup by max since we are acting on finite, normed vector spaces. It is equivalent to seek for the maximimum of a function u or instead seek for the minimum of the function $-u$.

Assuming that problem **??** we are considering has at least one solution, which we denote by $(\hat{u}, \hat{p}) \in X \times Y$, then $\hat{u}$ as the solution to the primal-problem satisfies

$$K\hat{u} \in \partial F^*(\hat{p}), \tag{3.6}$$

and $\hat{p}$ as the solution of the dual-problem satisfies

$$-(K^*\hat{p}) \in \partial G(\hat{u}). \tag{3.7}$$

Proof If we define a function $L(u,p) := \langle K^*p, u\rangle - F^*(p) + G(u)$ in the sense of equation **??**, we have

$$\max_{p\in Y} \min_{u\in X} L(u,p)$$

Now, fixing the maximal value p and denote this by $\hat{p}$, we get the following minimzation problem

$$\min_{u\in X} L(u,\hat{p}).$$

According to Proposition **??** this is equivalent to $0 \in \partial\, L(\hat{u},\hat{p})$ if $\hat{u}$ is the minimizer of the optimization problem. From this we observe equation **??**, since

$$0 \in \partial\left(\langle K^*\hat{p}, \hat{u}\rangle - F^*(\hat{p}) + G(\hat{u})\right) \Longleftrightarrow 0 \in K^*\hat{p} + \partial\, G(\hat{u}) \Longleftrightarrow -K^*\hat{p} \in \partial\, G(\hat{u}).$$

Doing the same by fixing u, denoted by $\hat{u}$ we observe

$$\max_{p\in Y} L(\hat{u},p) \Longleftrightarrow \min_{p\in Y} -L(\hat{u},p).$$

Again, using Proposition **??** we get

$$0 \in \partial\, -\left(\langle \hat{p}, K\hat{u}\rangle - F^*(\hat{p}) + G(\hat{u})\right) \Longleftrightarrow 0 \in -K\hat{u} + \partial\, F^*(\hat{p}) \Longleftrightarrow K\hat{u} \in F^*(\hat{p}),$$

which proves Equation **??**. ■

3.2 A First-Order Primal-Dual Algorithm

Our goal is to solve the saddle-point problem **??**. Therefore, one finds in a series of paper the celebrated (fast) first-order Primal-Dual Algorithm. According to [**?**] the idea of this method for solving saddle-point problems goes back to Arrow and Hurwicz, see also [**?**]. For that reason these primal-dual approaches are sometimes also called Arrow-Hurwicz methods. The first time this algorithm was stated in a framework was probably in [**?**]. The general idea of the proposed algorithm is to do a gradient descent in u, since this is the variable of the minimzation problem. And do, simultaneously, a gradient ascent in p, because this is the variable of the maximization problem. Choosing time-steps $\sigma, \tau > 0$ one gets

$$\begin{aligned} p^{n+1} &= (\mathrm{Id} + \sigma\, \partial\, F^*)^{-1}(p^n + \sigma\, K u^n) \\ u^{n+1} &= (\mathrm{Id} + \tau\, \partial\, G)^{-1}(u^n - \tau\, K^* p^{n+1}). \end{aligned}$$

The algorithm is proposed in a paper of Zhu and Chan in [**?**]. Unfortunatelly, there is no proof of convergence for this scheme. This makes the approach poor. But in 2009 Pock, Cremers, Bischof and Chambolle proposed a paper, that we also consider in Chapter 3, whose contribution was a provable extension of the above scheme. The idea here was to add an additional extrapolation step to the algorithm, as seen in line three of the primal-dual algorithm. In [**?**] Pock and Chambolle generalized this algorithm. They also proposed some variations of the algorithm itself. Depending on the properties of the corresponding functions F^* and G one can derive a better convergence rate. We will not provide details and just make use of two algorithms. In Chapter 4, we apply the two versions to different problems in imaging. Further, we will not provide a proof of convergence for these algorithms. For details we refer to [**?**] and [**?**]. Then the general fast primal-dual algorithm is as follows

Algorithm 3.1 (First-Order Primal-Dual Algorithm) *Choose $(u^0, p^0) \in X \times Y$ and let $\bar{u}^0 = u^0$. Further let $\tau, \sigma > 0$ with $\sigma\tau L^2 \leq 1$ and $\theta \in [0, 1]$. Then, we let for each $n \geq 0$*

$$\begin{cases} p^{n+1} = (\mathrm{Id} + \sigma\, \partial\, F^*)^{-1}(p^n + \sigma\, K\bar{u}^n) \\ u^{n+1} = (\mathrm{Id} + \tau\, \partial\, G)^{-1}(u^n - \tau\, K^* p^{n+1}) \\ \bar{u}^{n+1} = u^{n+1} + \theta(u^{n+1} - u^n). \end{cases} \tag{3.8}$$

Additionally, let us introduce the primal-dual gap, which is strongly related to the weak and strong duality theorems (found for instance in [**?**]. The primal-dual gap will be a part of the convergences theorem **??**.

Definition 3.2 (Primal-Dual Gap) *Let $u \in X$, $p \in Y$ be the variables of the optimization problem in Equation **??**. Then we define the primal-dual gap of this problem by*

$$\mathcal{G}(u, p) = \max_{\tilde{p} \in Y} \langle \tilde{p}, Ku \rangle - F^*(\tilde{p}) + G(u) - \min_{\tilde{u} \in X} \langle p, K\tilde{u} \rangle - F^*(p) + G(\tilde{u}), \tag{3.9}$$

which has the property that $\mathcal{G}(u, p) \geq 0$ for all u, p and equality only holds if and only if (u, p) is a saddle-point. If $\hat{p}$ is a solution of the maximization problem and $\hat{u}$ a solution of the minimization problem the following inequality holds:

$$\mathcal{G}(u, p) \geq \langle \hat{p}, Ku \rangle - F^*(\hat{p}) + G(u) - \langle p, K\hat{u} \rangle - F^*(p) + G(\hat{u}) \geq 0 \tag{3.10}$$

For this particular algorithm one can find a convergence theorem in [**?**], which we provide without a proof.

Theorem 3.3 *Let $L = ||K||$ and assume Equation **??** has a saddle-point $(\hat{u}, \hat{p})$. Choose $\theta = 1, \tau, \sigma, L^2 < 1$, and let $(u^n, \bar{u}^n, p^n)$ be defined by **??**. Then*

1. *For any n,*

$$\frac{||p^n - \hat{p}||^2}{2\sigma} + \frac{||u^n - \hat{u}||^2}{2\tau} \leq C\left(\frac{||p^0 - \hat{p}||^2}{2\sigma} + \frac{||u^0 - \hat{u}||^2}{2\tau}\right),$$

where the constant $C \leq (1 - \tau\sigma L^2)^{-1}$.

2. *If we let* $u^N = \left(\frac{\sum_{n=1}^{N} u^n}{N}\right)$ *and* $p^N = \left(\frac{\sum_{n=1}^{N} p^n}{N}\right)$, *for any bounded* $B_1 \times B_2 \subset X \times Y$ *the restricted primal-dual gap has the following bound:*

$$\mathcal{G}_{B_1 \times B_2}(u^N, p^N) \leq \frac{D(B_1, B_2)}{N},$$

where

$$D(B_1, B_2) = \sup_{(u,p) \in B_1 \times B_2} \frac{||u - u^0||^2}{2\tau} + \frac{||p - p^0||^2}{2\sigma}.$$

Moreover, the weak cluster points of (u^N, p^N) *are saddle-points of* **??**.

3. *If the dimension of the spaces* X *and* Y *is finite, then there exists a saddle-point* (u^*, p^*), *such that* $u^n \longrightarrow u^*$ *and* $p^n \longrightarrow p^*$.

Remark 3.4 *What this theorem states is, that one needs to choose* τ, σ *carefully by initializing the algorithm. As long as the inequality* $\tau\sigma L^2 < 1$ *holds, convergence is guaranteed. The two parameters* τ, σ *are also sometimes called time-steps. The better the choice for these beforhand, the faster the algorithm converges. Of course, it is not the best way having an algorithm which is dependent on manual choices, but on the other hand two parameter are highly controllable and one gets fast convergence. Other methods are almost as fast as the primal-dual algorithm, but depend on a couple of parameters, or they are independent of parameter choices and have a slow convergence rate.*

As mentioned, sometimes this method for solving saddle-point problems is also called Arrow-Hurwicz methods. To derive their original algorithm one would need to choose $\theta = 0$. We will not consider this case in our computations. The convergence rate for the primal-dual algorithm is $\mathcal{O}(\frac{1}{N})$. In the case that one function, F^* or G, is uniformly convex one gets:

Algorithm 3.5 (F^* or G are uniformly convex) *Choose* $(u^0, p^0) \in X \times Y$ *and let* $\bar{u}^0 = u^0$. *Further let* $\tau_0, \sigma_0, \gamma > 0$ *with* $\sigma\tau L^2 \leq 1$. *Then, we let for each* $n \geq 0$

$$\begin{cases} p^{n+1} = (\mathrm{Id} + \sigma_n \partial F^*)^{-1}(p^n + \sigma_n K \bar{u}^n) \\ u^{n+1} = (\mathrm{Id} + \tau_n \partial G)^{-1}(u^n - \tau_n K^* p^{n+1}) \\ \theta_n = \frac{1}{\sqrt{1+2\gamma\tau_n}},\ \tau_{n+1} = \theta_n \tau_n,\ \sigma_{n+1} = \frac{\sigma_n}{\theta_n} \bar{u}^{n+1} = u^{n+1} + \theta_n(u^{n+1} - u^n). \end{cases} \tag{3.11}$$

This leads to a convergence rate of $\mathcal{O}(\frac{1}{N})$. Once parallelized, these algorithms will run in real-time. Fortunatelly, both are highly parallelizable on a GPU using the CUDA framework. In Chapter 4 we provide in-depth details of this.

The computations in each line are straightforward. One has to compute sums of vectors and scaled vectors. Since, we will set $K = \nabla$ and for that its conjugate is $-\mathrm{div}$ or equivalently $-\nabla^T$, calculating the operator is also easy. The main work needs to be done to find the proximity operators for G and F^*, respectively. They vary within different models. For that, we need to find these operators for each and every model.

3.3 Discrete Setting

In our discrete setting we consider a regular pixel grid of size $N \times M$ and set

$$\mathcal{G} = \Big\{(i,j) : i = 1, ..., N \text{ and } j = 1, ..., M\Big\}. \tag{3.12}$$

where the indices (i, j) denote the discrete locations in the pixel grid. And we define our image $u \in X : \mathcal{G} \longrightarrow \mathbb{R}$ where $X \in \mathbb{R}^{N \cdot M}$ is a finite dimensional vector space equipped with the standard inner product

$$\langle u, v \rangle_X = u^T v = \sum_{i=1}^{N} \sum_{j=1}^{M} u_{i,j} v_{i,j}, \;\; u, v \in X, \tag{3.13}$$

and the standard euclidean norm

$$||u||_2 = \left(u^T u\right)^{\frac{1}{2}} = \langle u, u \rangle^{\frac{1}{2}}, \;\; u \in X.$$

Furthermore, let $Y = X \times X$ be the dual space of X equipped with the inner product defined by

$$\langle p, q \rangle_Y = p^T q = \sum_{i=1}^{N} \sum_{j=1}^{M} p^1_{i,j} q^1_{i,j} + p^2_{i,j} q^2_{i,j},$$

with $p_{i,j} = \left(p^1_{i,j}, p^2_{i,j}\right)^T, q_{i,j} = \left(q^1_{i,j}, q^2_{i,j}\right)^T \in Y$ and also equipped with the euclidean norm.

For simplicity we let our normed vector space X be in $\mathbb{R}^n$ and swap the superscript $N \cdot M$ to n. If the exact space size is of importance, we will make this clear. Further, we define $\mathbb{R}_\infty = \mathbb{R} \cup \{\infty\}$ as the space of the extended real values and $\mathbb{R}_+ = [0, +\infty)$ as the space of all positive real values including zero.

Because the total variation is associated with the gradient operator, we want to give some discretization for this operator. As we saw in Equation **??**, the gradient ∇ becomes $-$div by swapping it from our function u to the testfunction φ. This important relation is used in the following and for that we further define the discrete divergence operator.

Definition 3.6 (Discrete gradient operator) *We define the discrete gradient of a vector $u \in X$ by $\nabla u = ((\partial_i u)_{i,j}, (\partial_j u)_{i,j})^T$ using forward differences with Neumann boundary conditions, i.e*

$$(\partial_i u)_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases} \tag{3.14}$$

$$(\partial_j u)_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < M \\ 0 & \text{if } j = M \end{cases} \tag{3.15}$$

We choose $\mathrm{div} : Y \longrightarrow X$ to be the discrete divergence operator, which relates to ∇ with $-\nabla^* = \mathrm{div}$. That is, for every $p \in Y$ and $u \in X$,

$$\langle \nabla u, p \rangle_Y = \langle u, \nabla^* p \rangle_X = -\langle u, \mathrm{div}\, p \rangle_X$$

as in Equation **??**.

Definition 3.7 (Discrete divergence operator) *We define the discrete divergence of a vector $p \in Y$ by $\nabla^T p = \partial_i p^1_{i,j} + \partial_j p^2_{i,j}$ using backward differences with Dirichlet boundary conditions, i.e*

$$(\partial_i p^1)_{i,j} = \begin{cases} p^1_{i,j} - p^1_{i-1,j} & \text{if } 1 < i < N \\ p^1_{i,j} & \text{if } i = 1 \\ -p^1_{i-1,j} & \text{if } i = N \end{cases} \tag{3.16}$$

$$(\partial_j p^2)_{i,j} = \begin{cases} p^2_{i,j} - p^2_{i,j-1} & \text{if } 1 < j < M \\ p^2_{i,j} & \text{if } j = 1 \\ -p^2_{i,j-1} & \text{if } j = M \end{cases} \tag{3.17}$$

Proposition 3.8 (Bound on the norm of ∇) *The bound on the norm of the proposed discrete linear operators is given by*

$$L^2 = ||\nabla|| = ||\mathrm{div}|| \leq 8. \tag{3.18}$$

Proof With the definitions of **??**, **??** and **??** we get

$$\begin{aligned} L &= ||\nabla|| = \max_{||x||_X \leq 1} ||\nabla x||_Y \\ &= ||\mathrm{div}|| = \max_{||p||_Y \leq 1} ||\mathrm{div}\, p||_X \\ &= \max_{||p||_Y \leq 1} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(p^1_{i,j} - p^1_{i-1,j} + p^2_{i,j} - p^2_{i,j-1}\right)^2 \\ &\leq 4 \max_{||p||_Y \leq 1} \sum_{i=1}^{N} \sum_{j=1}^{M} (p^1_{i,j})^2 + (p^1_{i-1,j})^2 + (p^2_{i,j})^2 + (p^2_{i,j-1})^2 \\ &\leq 8 \max_{||p||_Y \leq 1} ||p||_Y^2 = 8. \end{aligned} \tag{3.19}$$

■

3.4 The ROF Model

Since, there exists a couple of models to approximate an input image g by a function u, we want to make some conventions about this. Our models are based on the idea that a given image g consists of two things: data and noise. This can be expressed by

$$g = g_d + g_n,$$

where g_d denotes the (actual) data of the image and g_n the noise which should be removed.

An example model to efficiently remove Gaussian noise from an image input image g would be the so called ROF model. Before introducing this model in detail, let us state one general property of all our underlying models. What they all have in common is the idea how they are set up:

$$\text{Model} = \text{Data Fidelity Term} + \text{Regularizer Term}.$$

The data fidelity term assures that the approximation u is as close to the input image g as possible. For instance, one can set $G(u) = \frac{1}{2}||u-g||_2^2$ having the quadratic, euclidean distance as the data fidelity term. As a quadratic norm function this term would be convex. For the regularizer we will see that we find convex terms which are easy to handle, but we also find highly non-convex regularizers like in the Mumford-Shah Functional, discussed in Chapter 3.

The first model we consider in this thesis is the ROF Model, named after Leonid I. Rudin, Stanley Osher and Emad Fatemi. They first proposed this model in 1992 in [**?**]. It is the prototype when talking about variational methods in image processing. For this we first define two important norms, which will appear. We define the discrete isotropic total variation norm by

$$||\nabla u||_1 = \sum_{i=1}^{N}\sum_{j=1}^{M} |(\nabla u)_{i,j}|, \text{ where } |(\nabla u)_{i,j}| = \sqrt{((\nabla u)_{i,j}^1)^2 + ((\nabla u)_{i,j})^2}. \tag{3.20}$$

Additionally, we define the discrete maximum (or l_∞) norm by

$$||p||_\infty = \max_{i,j} |p_{i,j}|, \text{ where } |p_{i,j}| = \sqrt{(p_{i,j}^1)^2 + (p_{i,j}^2)^2}. \tag{3.21}$$

Definition 3.9 (ROF Model) *Let $\Omega \in \mathbb{R}^d$ be the d-dimensional image domain, $u \in W_1^1(\Omega)$ and $g \in L^1(\Omega)$ a (noisy) input image. Then the ROF model is defined as the variational problem*

$$\min_u E_{ROF}(u) = \min_u \mathrm{TV}(u) + \frac{\lambda}{2}\int_\Omega |u-g|^2\,dx = \min_{u\in X} \int_\Omega |\nabla u|\,dx + \frac{\lambda}{2}\int_\Omega |u-g|^2\,dx. \tag{3.22}$$

The appearing parameter λ is used to model the tradeoff between the regularizer, namely the total variation, and the data fidelity term. Having a larger value for λ one gets an approximation u which is closer to the input image g. Taking in account the total variation, the model is able to preserve edges in its output. For that, a smaller λ favors sharper edges, but is not as close to the input image g. Reformulation of Equation **??** into a discrete setting leads to:

$$\min_{u\in X} E_{ROF}(u) = \min_{u\in X} ||\nabla u||_1 + \frac{\lambda}{2}||u-g||_2^2, \tag{3.23}$$

with $u \in X$ being the unknown approximation and $g \in X$ the given noisy data.

Remark 3.10 *In some literature, for instance [?], one finds a multiplicative factor h^2 in Equation ??. This factor is due to discretization. Since we assume our image domain Ω having its pixel values as discrete locations, we do not make use of the additional factor in none of our models. Note, that it only rescales the optimization problem and does not change the solution.*

3.4.1 ROF Model as Saddle-Point Problem

Let us now rewrite the minimization problem in Equation ?? to derive the saddle-point formulation from Section ??. For that we define $F(\nabla u) := ||\nabla u||_1$ to be the total variation and $G(u) := \frac{\lambda}{2}||u-g||_2^2$. Hence, we are facing the following optimization problem

$$\min_{u\in X} F(\nabla u) + G(u) = \min_{u\in X} ||\nabla u||_1 + \frac{\lambda}{2}||u-g||_2^2. \tag{3.24}$$

Applying the Legendre-Fenchel conjugate on F one has $F(\nabla u) = \sup_{p\in Y}\langle p, \nabla u\rangle_Y - F^*(p)$, and we observe the saddle-point problem

$$\min_{u\in X}\max_{p\in Y} \langle p, \nabla u\rangle_X - F^*(p) + G(u) = \min_{u\in X}\max_{p\in Y} -\langle \nabla^T p, u\rangle_X - F^*(p) + G(u). \tag{3.25}$$

Now, it remains to show how $F^*(p)$ looks like. From Example ?? 2) we know, that for any norm the conjugate is given by

$$F^*(p) = \begin{cases} 0 & \text{if } ||p||_* \leq 1, \\ \infty & \text{else,} \end{cases} \tag{3.26}$$

or equivalently $F^* = \delta_P(p)$ for a set P given by

$$P = \big\{p \in Y : ||p||_\infty \leq 1\big\}. \tag{3.27}$$

Here, we used that the conjugate of the l_1 norm is the l_∞ norm. This leads to

$$\min_{u\in X}\max_{p\in Y} \langle p, \nabla u\rangle_Y + \frac{\lambda}{2}||u-g||_2^2 - \delta_P(p). \tag{3.28}$$

By Example ?? 3. the conjugate of the function G is given by $G^*(p) = \frac{\lambda}{2}||p-g||_2^2$ since the conjugate of the euclidean norm is itself the euclidean norm. And we already know, that $F^*(p) = \delta_P(p)$. Plugging these two equations into ?? and having $K^* = \nabla^T$ we get

$$\begin{aligned} \max_{p\in Y} -(G^*(-K^*p) + F^*(p)) &= \max_{p\in Y} -\left(\frac{\lambda}{2}||-\nabla^T p - g||_2^2 + \delta_P(p)\right) \\ &= \max_{p\in Y} -\left(\frac{\lambda}{2}||\nabla^T p + g||_2^2 + \delta_P(p)\right) \end{aligned} \tag{3.29}$$

which is the dual of the ROF model. In [?] one could find another notation for the dual ROF model, namely

$$\max_{p\in Y} -\left(\frac{1}{2\lambda}||\nabla^T p||_2^2 + \langle g, \nabla^T p\rangle_X + \delta_P(p)\right).$$

This is actually equivalent to our formulation. First note that the parameter λ in Equation **??** can be swapped from the data fidelity part to the regularizer by $\frac{1}{\lambda}$. This scaling factor does not change the energy at all. To derive the first two terms in this notation one just needs to compute $||\nabla^T p + g||_2^2 = ||\nabla^T p||_2^2 + ||g||_2^2 + 2\langle \nabla^T p, g\rangle$. The factor two vanishes by multiplying it with $\frac{1}{2}$. And the term $||g||_2^2$ as a constant factor can be crossed out since it only shifts the energy. This shows the equivalence of the two stated formulations.

3.4.2 The Proximity Operators of the ROF Model

In Algorithm **??** we saw that solving saddle-point problems can be done in three lines of pseudo code. One important step in this scheme was to calculate the proximity operator for $(\mathrm{Id} + \sigma\,\partial\, F^*)^{-1}$ and $(\mathrm{Id} + \tau\,\partial\, G)^{-1}$), respectively. As mentioned before, we need to find the corresponding operator for each model. Within the ROF Model ones has $G(u) = \frac{\lambda}{2}||u - g||_2^2$. As we already saw we further have $F^*(p) = \delta_P(p)$. Applying Equation **??** to F^* we get

$$(\mathrm{Id} + \sigma\,\partial\, F^*)^{-1}(\tilde{p}) = \min_{p\in P} \frac{||p - \tilde{p}||_2^2}{2} + \sigma\,\delta_P(p) = \min_{p\in P} \frac{||p - \tilde{p}||_2^2}{2}.$$

This is nothing but the euclidean projection of a vector $\tilde{p} \notin P$ onto the convex set P. From Example **??** 1) we have

$$(\mathrm{Id} + \sigma\,\partial\, F^*)^{-1}(\tilde{p}) = P_{l_2}(\tilde{p}) = p \Longleftrightarrow p_{i,j} \frac{\tilde{p}_{i,j}}{\max(1, |\tilde{p}_{i,j}|)}, \tag{3.30}$$

which holds for all $i = 1, ..., N, j = 1, ..., M$. Indeed, this projection goes pointwise and can easily be implemented. It remains to compute the proximity operator for our function G. Here, we get

$$(\mathrm{Id} + \tau\,\partial\, G)^{-1}(\tilde{u}) = \min_{u\in X} \frac{||u - \tilde{u}||_2^2}{2} + \frac{\tau\lambda}{2}||u - g||_2^2 = \min_{u\in X} \mathcal{L}(u)$$

where we define $\mathcal{L}(u) := \frac{||u-\tilde{u}||_2^2}{2} + \frac{\tau\lambda}{2}||u - g||_2^2$. Then this minimization problem is equivalent to compute $\nabla\mathcal{L}(u) = 0$. It follows

$$\nabla\mathcal{L}(u) = (u - \tilde{u}) + \tau\lambda(u - g) = 0 \Longleftrightarrow (1 + \tau\lambda)u = \tilde{u} + \tau\lambda g \Longleftrightarrow u = \frac{\tilde{u} + \tau\lambda g}{1 + \tau\lambda}.$$

Then, again pointwise for all $i = 1, ..., N$ and $j = 1, ..., M$ we have

$$(\mathrm{Id} + \tau\,\partial\, G)^{-1}(\tilde{u}) = u \Longleftrightarrow u_{i,j} = \frac{\tilde{u}_{i,j} + \tau\lambda g}{1 + \tau\sigma} \tag{3.31}$$

Now, that everything is defined and set perfectly one can implement the ROF model. This algorithm is massively parallelizable. We used in our framework C++ and CUDA respectively. We will discuss implementation issues and applications of the ROF Model in detail in Chapter 4.

3.5 The TVL1 Model

In the previous section we discussed the ROF Model. This model is sometimes also known as the TVL2 Model. The reason for that is, that we set the total variation term as the regularizer and use a L^2 norm in the data fidelity term. The idea now is to replace the norm in the data term with the L^1 norm. This norm is more robust in removing the so called salt and pepper noise, meaning that it removes single pixels with extrem values white or black. The model looks as follows:

Definition 3.11 (The TVL1 Model) *Let $\Omega \in \mathbb{R}^d$ be the d-dimensional image domain, $u \in W_1^1(\Omega)$ the approximation and $g \in L^1(\Omega)$ a (noisy) input image. Then the TVL1 Model is defined as the variational problem*

$$\min_u E_{TVL1}(u) = \min_u \mathrm{TV}(u) + \lambda \int_\Omega |u-g|\,dx = \min_{u\in X} \int_\Omega |\nabla u|\,dx + \lambda \int_\Omega |u-g|\,dx. \quad (3.32)$$

Note, that there is also the parameter λ to handle the tradeoff between both terms. If we reformulate this into a discrete version we have

$$\min_{u\in X} E_{TVL1}(u) = \min_{u\in X} ||\nabla u||_1 + \lambda||u-g||_1, \quad (3.33)$$

where again u is the approximation of an input image g.

3.5.1 TVL1 as Saddle-Point Problem

As we did in the previous section, we can formulate this minimization problem into a saddle-point problem to apply the Primal-Dual Algorithm. Let us first state that our function $F(\nabla u)$ remains the same. We only have a change in G. Then the primal problem of the TVL1 Model becomes

$$\min_{u\in X}\ F(\nabla u) + G(u) = \min_{u\in X}\ ||\nabla u||_1 + \lambda||u-g||_1. \quad (3.34)$$

Again using the Legendre-Fenchel conjugate for F - as in subsection **??** we obtain

$$\min_{u\in X}\max_{p\in Y} \langle p, \nabla u\rangle_X - F^*(p) + G(u) = \min_{u\in X}\max_{p\in Y} -\langle \nabla^T p, u\rangle_X - F^*(p) + G(u). \quad (3.35)$$

From Equation **??** we already know how F^* looks like. Writing it as the indicator function we got $F^* = \delta_P(p)$ with the set P of Equation **??**. Then the primal-dual problem becomes

$$\min_{u\in X}\max_{p\in Y} \langle p, \nabla u\rangle_Y + \lambda||u-g||_1 - \delta_P(p). \quad (3.36)$$

We find the conjugate of the function $G(u) = \lambda||u-g||_1$ is given by

$$G^*(q) = \begin{cases} 0 & \text{if } ||q||_* \le \lambda, \\ \infty & \text{else,} \end{cases} \quad (3.37)$$

which means nothing but $G^*(q) = \delta_Q(q)$ for a set

$$Q = \big\{q \in Y : ||q||_\infty \le \lambda\big\}. \quad (3.38)$$

Proof To derive this representation of the conjugate function we set $z = u - g$, which is equivalent to $u = z + g$. Then with the definition of the Legendre-Fenchel conjugate we get

$$\begin{aligned} G^*(q) = \sup_{u\in X} \langle q, u\rangle - G(u) \Longleftrightarrow G^*(q) &= \sup_{z\in X} \langle q, z+g\rangle - G(z+g) \\ &= \sup_{z\in X} \langle q, z+g\rangle - \lambda||z||_1 \\ &= \sup_{z\in X} \frac{1}{\lambda}\big(\langle q, z\rangle + \underbrace{\langle p, g\rangle}_{=\text{const}}\big) - ||z||_1 \\ &= \sup_{z\in X} \langle \frac{1}{\lambda} q, z\rangle - ||z||_1. \end{aligned} \tag{3.39}$$

Since we are facing to compute the Legendre-Fenchel conjugate of the l^1 norm we already know from Example **??** 2. that

$$G^*(q) = \begin{cases} 0 & \text{if } ||\frac{1}{\lambda} q||_* \leq 1, \\ \infty & \text{else,} \end{cases} \Longleftrightarrow G^*(q) = \begin{cases} 0 & \text{if } ||q||_* \leq \lambda, \\ \infty & \text{else.} \end{cases}$$

■

We obtain the dual formulation of the TVL1 Model as

$$\max_{p\in Y} \; -(G^*(-K^*p) + F^*(p)) = \max_{p\in Y} -\Big(\delta_Q(\nabla^T q) + \delta_P(p)\Big). \tag{3.40}$$

The Proximity Operators of the TVL1 Model

For the implementation of the Primal-Dual Algorithm we need the proximity operators of the TVL1 Model. Fortunatelly, the operator for the function F^* remains the same as in Subsection **??** since the functions are the same. We get again

$$(\mathrm{Id} + \sigma\, \partial\, F^*)^{-1}(\tilde{p}) = P_{l_2}(\tilde{p}) = p \Longleftrightarrow p_{i,j} \frac{\tilde{p}_{i,j}}{\max(1, |\tilde{p}_{i,j}|)}, \tag{3.41}$$

for all $i = 1, ..., N, j = 1, ..., M$.

To compute the proximity operator of the function G we apply again Definition **??**. Then we have

$$(\mathrm{Id} + \tau\, \partial\, G)^{-1}(\tilde{u}) = \min_{u\in X} \frac{||u - \tilde{u}||_2^2}{2} + \lambda\tau||u - g||_1.$$

If we define $\mathcal{L}(u) = \frac{||u-\tilde{u}||_2^2}{2} + \lambda\tau||u||_1$ then the minimization problem $\min_{u\in X} \mathcal{L}(u)$ is equivalent to $\nabla\mathcal{L}(u) = 0$. We already saw that the l^1 norm is not differentiable everywhere and for that non-smooth. To compute the gradient we need need the partial derivatives for each $i = 1, ..., N$. Let us first take a look at the i-th row of $\nabla\mathcal{L}(u)$:

$$\partial_i(\mathcal{L}(u)) = u_i - \tilde{u}_i + \tau\lambda\partial_i(||u-g||_1) = u_i - \tilde{u}_i + \tau\lambda\partial_i(|u_1 - g_1| + ... + |u_i - g_i| + ... + |u_n - g_n|).$$

With that we see that we need to compute the subgradient of the absolute value function in the i-th line of the above equation. From Example **??** 1. we have

$$y_i = \begin{cases} 1 & \text{if } u_i - g_i > 0, \\ -1 & \text{if } u_i - g_i < 0, \\ [-1,1] & \text{if } u_i - g_i = 0 \Longleftrightarrow u_i = g_i, \end{cases}$$

where y_i is the subgradient of the i-th row. Let us check all three cases:

1. Let $y_i = 1$. Then we obtain for the i-th row

   $$u_i - \tilde{u}_i + \tau\lambda = 0 \Longleftrightarrow u_i = \tilde{u}_i - \tau\lambda.$$

   From $u - g_i > 0$ we know that this equation holds for $\tilde{u}_i - \tau\lambda - g_i > 0$, which is then equivalent to $\tilde{u}_i - g_i > \tau\lambda$.

2. Now, let $y_i = -1$. The we get in row i

   $$u_i - \tilde{u}_i - \tau\lambda = 0 \Longleftrightarrow u_i = \tilde{u}_i + \tau\lambda.$$

   Rewriting the constraint we obtain $\tilde{u}_i + \tau\lambda - g_i > 0$. From this it follows $\tilde{u} - g_i > \tau\lambda$.

3. Finally, $y_i \in [-1, 1]$. Since, we know that $u_i = g_i$ we get

   $$g_i - \tilde{u}_i + \tau\lambda y_i = 0 \Longleftrightarrow \tilde{u}_i - g_i = \tau\lambda y_i.$$

   We apply the absolute value function to each side of the equation, then with $|y_i| \leq 1$ we have

   $$|\tilde{u}_i - g_i| = |\tau\lambda y_i| \leq \tau\lambda.$$

Overall, we have pointwise

$$u = (\mathrm{Id} + \tau\,\partial\, G)^{-1}(\tilde{u}) \Longleftrightarrow u_{i,j} = \begin{cases} \tilde{u}_{i,j} - \tau\lambda & \text{if } \tilde{u}_{i,j} - g_{i,j} > \tau\lambda, \\ \tilde{u}_{i,j} + \tau\lambda & \text{if } \tilde{u}_{i,j} - g_{i,j} < -\tau\lambda, \\ g_{i,j} & \text{if } |\tilde{u}_{i,j} - g_{i,j}| \leq \tau\lambda. \end{cases} \tag{3.42}$$

Note, that we used the discrete locations of the pixel (i, j) in this notation instead of the row-wise notation i from above. Of course, this does not change any of the solutions.

3.6 The Mumford-Shah Functional

We saw that solving the ROF Model and TVL1 Model, respectively, depended mainly on finding the right choices of the convex conjugates and the proximity operator. Now, we introduce a highly non-convex functional which is more difficult to solve. The problem of the proposed method in this section is, that there is - to date - no proof that it convergences to a global minimum. Even though, one can see that it yields to high quality solutions. But, of course one wants to be sure to compute the global optimum. Therefore, we discuss in the next chapter a method to minimize the Mumford-Shah functional optimally. For now, let us first give the definition of this functional.

Definition 3.12 (The Mumford-Shah Functional [1]) *Let $\Omega \subset \mathbb{R}^2$ be a rectangular image domain. In order to approximate an input image $g : \Omega \longrightarrow \mathbb{R}$ in terms of a piecewise smooth function $u : \Omega \longrightarrow \mathbb{R}$, the Mumford-Shah Functional is given by*

$$E(u) = \int_\Omega (u-g)^2 dx + R(u) = \int_\Omega (u-g)^2 dx + \lambda \int_{\Omega \setminus K} |\nabla u|^2 dx + \nu |K|, \tag{3.43}$$

where $\nu, \lambda > 0$ are weighting parameters, $K = K_1 \cup ... \cup K_N$ and $|K|$ denotes the length of the curves in K.

Remark 3.13 *In this section we follow the represenation of the Mumford-Shah Functional of [1]. This is equivalent to Equation 2.1, but has some advantages for our notation. In Chapter 3 we will state a second notation.*

This functional differs from the ones in the previous sections. The first term, the data fidelity term, remains the same as in the ROF or TVL1 Model. The approximation u should be as close to the input image g as possible. The regularizer consists of two terms. The first term of this kind of regularizer, also known as Mumford-Shah regularizer, uses again the gradient, but not in the set Ω itself. Instead it states that the approximation u is not allowed to change too much in sets $\Omega \setminus K_i$. We call K_i the discontinouty sets (or jump sets) for all $i = 1, ..., N$ and curves u_i. The gradient is only taken into account in these regions if they are smooth enough. These discontinouties of the sets K_i are also measured and taken into account in the energy $E(u)$. This means, that all curves K_i should be regular in the sense of measure theory. Here, we also find two weighting parameters ν and λ. Where λ handles the tradeoff between the first two terms as in the ROF or TVL1 Model, ν controls the length of the discontinouty sets. A smaller ν yields to a smoother image, where a higher ν leads to sharper edges in our images. The parameter λ yet plays another important role. If one chooses a λ small enough, then our model is also called piecewise-smooth Mumford-Shah Model. On the other hand, in the limiting case $\lambda \longrightarrow \infty$, we can only attain a minimum if we set $\nabla u = 0$ in $\Omega \setminus K$. Then the model is known as the piecewise-constant Mumford-Shah Model. In [1] Strekalovskiy and Cremers proposed to rewrite this functional in a discrete setting by first defining the discrete regularizer function by

$$R_{MS}(u) = \min(\lambda ||u||_2^2, \nu). \tag{3.44}$$

Then the discrete Mumford-Shah Model can be expressed by

$$\min_{u \in X} E_{MS}(u) = \min_{u \in X} ||u-g||_2^2 + R_{MS}(\nabla u). \tag{3.45}$$

According to [1] the idea behind this formulation is to model the discontinouity set K explicitly in terms of the function u. This means, that K is the set of all points where the minimum in 2.2 attains ν. In other words, if the gradient ∇u is large enough we have for the explicit set K_{MS}:

$$K_{MS} = \left\{ (i,j) \in \Omega : ||\nabla u_{i,j}||_2^2 \geq \sqrt{\frac{\nu}{\lambda}} \right\}. \tag{3.46}$$

We can check easy that for a point $(i,j) \in K_{MS}$ we observe

$$R_{MS}(\nabla u_{i,j}) = \min(\underbrace{\lambda||\nabla u_{i,j}||_2^2}_{\geq \lambda\sqrt{\frac{\nu}{\lambda}}^2 = \nu}, \nu) = \nu$$

and if $(i,j) \notin K_{MS}$ we have

$$R_{MS}(\nabla u_{i,j}) = \min(\underbrace{\lambda||\nabla u_{i,j}||_2^2}_{< \lambda\sqrt{\frac{\nu}{\lambda}}^2 = \nu}, \nu) = \lambda||\nabla u_{i,j}||_2^2.$$

Remark 3.14 *In the piecewise-constant case, where $\lambda \longrightarrow \infty$, Equation 2.2 changes to*

$$R_{MS}(u) = \begin{cases} \nu & \text{if } u \neq 0, \\ 0 & \text{else.} \end{cases} \tag{3.47}$$

3.6.1 Mumford-Shah as Saddle-Point Problem

Again, we try to formulate the Mumford-Shah Model as a saddle-point problem to be able to apply the second primal-dual algorithm of Section **??**. In the sense of our notations from the previous section we have

$$\min_{u \in X} \; F(\nabla u) + G(u) = R_{MS}(\nabla u) + ||u - g||_2^2 \tag{3.48}$$

This is the primal formulation for the discrete Mumford-Shah Model. Applying now the Legendre-Fenchel conjugate on the Mumford-Shah regularizer R_{MS} we get the primal-dual formulation with

$$\min_{u \in X} \max_{p \in Y} \langle p, \nabla u \rangle_X - R_{MS}^*(p) + G(u). \tag{3.49}$$

As before, we want to compute the convex conjugate of R_{MS}. We have

$$R_{MS}^*(p) = \sup_{u \in X} \langle p, u \rangle - R_{MS}(u) = \sup_{p \in Y} \langle p, u \rangle - \min(\lambda||u||_2^2, \nu).$$

We need to distinguish both cases of our minimum function.

1. Assume that $\min(\lambda||u||_2^2, \nu) = \nu$. We get

   $$\sup_{u \in X} \langle p, u \rangle - \nu = S_Y(p) - \nu.$$

   where $S_Y(p)$ denotes the support function of Y for a point $p \in Y$. Clearly, if $p \neq 0$, then the supremum over all u is ∞. But if $p = 0$, this expression becomes

   $$S_Y(0) - \nu = -\nu \Longleftrightarrow -S_Y(0) + \nu = \nu.$$

2. On the other side, if we assume that $\min(\lambda||u||_2^2, \nu) = \lambda||u||_2^2$ we can apply Example **??** 2. and 4. Let $\hat{R}(u) = \frac{||u||_2^2}{2}$, then the Legendre-Fenchel conjugate of $R_{MS}(u) = 2\lambda\hat{R}(u)$ is $2\lambda\hat{R}^*(\frac{p}{2\lambda})$. We get

$$R_{MS}^*(p) = 2\lambda\hat{R}^*(\frac{p}{2\lambda}) = 2\lambda\frac{||\frac{p}{2\lambda}||_2^2}{2} = \frac{||p||_2^2}{4\lambda}. \tag{3.50}$$

If we now make use of Proposition **??**, we have that $p \in \partial R_{MS}(u)$ if and only if

$$\langle p, u\rangle - R_{MS}(u) = R_{MS}^*(p).$$

First, let us verify that this expression holds.

Proof "$\Longrightarrow$":
Let $p \in \partial R_{MS}(u) = 2\lambda u$, then this is equivalent to $p = 2\lambda u \Longleftrightarrow u = \frac{p}{2\lambda}$. Pluging this into the equation we get

$$\begin{aligned}\langle p, u\rangle - R_{MS}(u) &= \langle p, \frac{p}{2\lambda}\rangle - \lambda||\frac{p}{2\lambda}||_2^2 = \frac{1}{2\lambda}||p||_2^2 - \frac{\lambda}{4\lambda^2}||p||_2^2 \\ &= \frac{1}{\lambda}||p||_2^2 - \frac{1}{4\lambda}||p||_2^2 \\ &= \frac{1}{4\lambda}||p||_2^2 = R_{MS}^*(p). \end{aligned} \tag{3.51}$$

"$\Longleftarrow$":
Now, rewrite the equation to

$$R_{MS}(u) = \langle p, u\rangle - R_{MS}^*(p).$$

Taking the subdifferential on R_{MS} we observe

$$\partial R_{MS}(u) = \partial\Big(\langle p, u\rangle - R_{MS}^*(p)\Big) \Longleftrightarrow \lambda u = p \Longleftrightarrow p \in \lambda u = \partial R_{MS}(u).$$

■

Knowing that $p \in \partial R_{MS}(u)$ or $u = \frac{p}{2\lambda}$ respectively, we have that Equation 2.8 only holds if $\lambda||u||_2^2 \leq \nu$ or if $\lambda||\frac{p}{2\lambda}||_2^2 = \frac{||p||_2^2}{4\lambda} \leq \nu$ or equivalently $||p||_2 \leq \sqrt{2\lambda\nu}$.

Overall, the convex conjugate of the Mumford-Shah regularizer is given by

$$R_{MS}^*(p) = \begin{cases} \frac{||p||_2^2}{4\lambda}, & \text{if } ||p||_2 \in (0, \sqrt{2\lambda\nu}], \\ \nu, & \text{if } ||p||_2 = 0, \\ \infty, & \text{else.} \end{cases} \tag{3.52}$$

To derive the dual formulation of the Mumford-Shah Model we would just need to plug the derived definition of the convex conjugate into the dual problem.

$$\max_{p\in Y} -(G^*(-K^*p) + R^*_{MS}(p)) = \max_{p\in Y} -(||\nabla^T p - g||_2^2) + \frac{||p||_2^2}{2\lambda},$$

as long as $||p||_2 \in (0, \sqrt{\lambda\nu}]$. If $||p||_2 = 0$ we have

$$\max_{p\in Y} -(G^*(-K^*p) + R^*_{MS}(p)) = \max_{0=p\in Y} -(||\nabla^T p - g||_2^2) + \nu = -(g^T g + \nu) = -g^T g - \nu.$$

In all other cases we would observe

$$\max_{p\in Y} -(G^*(-K^*p) + R^*_{MS}(p)) = \max_{p\in Y} -(||\nabla^T p - g||_2^2 + \infty) = -\infty.$$

We are now interested in the proximity operators for our model, which we compute in the next subsection.

3.6.2 The Proximity Operators of the Mumford-Shah Model

Let us now compute the proximity operators of our proposed model. The operator for the function G can be adapted from Equation **??**, because our data fidelity term is the same as in the ROF Model. Then we have pointwise for all $i = 1, ..., N$ and $j = 1, ..., M$

$$(\mathrm{Id} + \tau\,\partial\, G)^{-1}(\tilde{u}) = u \Longleftrightarrow u_{i,j} = \frac{\tilde{u}_{i,j} + \tau\nu g}{1 + \tau\sigma}. \tag{3.53}$$

The more interesting case is to compute the proximity operator for $R^*_{MS}(p)$. Here, we have two possibilities to derive a formula for this. In [1] one can find this operator derived by using Moreau's Identity. We will show this case, too. But first we want to provide another way to derive the formula. We just computed $R^*_{MS}(p)$, so we easily apply the definition of the proximity operator. Since the third case of R^*_{MS} where the conjugate is ∞ is useless for us, we skip this case and first consider the case where $||p||_2 = 0$ which is equivalent to $p = 0$. For that we already see that if $||p||_2 = 0$ we set $p = 0$. If $||p||_2 \in (0, \sqrt{\lambda\nu}]$, then we get

$$\begin{aligned}\mathrm{prox}^{\sigma}_{R^*_{MS}(p)}(\tilde{p}) &= \min_p \frac{||p-\tilde{p}||_2^2}{2} + \sigma R^*_{MS}(p) \\ &= \min_p \frac{||p-\tilde{p}||_2^2}{2} + \sigma\frac{||p||_2^2}{4\lambda}\end{aligned}$$

Again, we define $\mathcal{L}(p) = \frac{||p-\tilde{p}||_2^2}{2} + \sigma\frac{||p||_2^2}{4\lambda}$ then $\min_p \mathcal{L}(p)$ is equivalent to solve $\nabla\mathcal{L}(p) = 0$. We have

$$\nabla\mathcal{L}(p) = p - \tilde{p} + \sigma\frac{p}{2\lambda} = 0 \Longleftrightarrow (1 + \frac{\sigma}{2\lambda})p = \tilde{p}.$$

Solving for p leads us to

$$p = \frac{\tilde{p}}{1 + \frac{\sigma}{2\lambda}} = \frac{2\lambda}{2\lambda + \sigma}\tilde{p}.$$

Now, that we know that $p = \frac{2\lambda}{2\lambda+\sigma}\tilde{p}$ we can plug this into the constraint to see for which condition on $\tilde{g}$ the computed value for p holds. Then with $\frac{||p||_2^2}{4\lambda} \leq \nu$ and $p \neq 0$ we have

$$
\begin{aligned}
\frac{||\frac{2\lambda}{2\lambda+\sigma}\tilde{p}||_2^2}{4\lambda} \leq \nu \quad &\Longleftrightarrow \quad \frac{4\lambda^2}{4\lambda(4\lambda+\sigma)^2}||\tilde{p}||_2^2 \leq \nu \\
&\Longleftrightarrow \quad \frac{4\lambda}{(4\lambda+\sigma)^2}||\tilde{p}||_2^2 \leq \nu \\
&\Longleftrightarrow \quad ||\tilde{p}||_2^2 \leq \frac{\nu}{4\lambda}(4\lambda+\sigma)^2 \\
&\Longleftrightarrow \quad ||\tilde{p}||_2 \leq \sqrt{\frac{\nu}{4\lambda}}\,(4\lambda+\sigma)
\end{aligned}
$$

!!! Then the proximity operator is defined by

$$
p = (\mathrm{Id} + \sigma\partial R_{MS}^*)^{-1}(\tilde{p}) \Longleftrightarrow p_{i,j} = \begin{cases} \dfrac{\lambda}{\lambda+\sigma}\tilde{p}, & \text{if } ||\tilde{p}||_2 \leq \sqrt{\frac{\nu}{\lambda}}\,(\lambda+\sigma), \\ 0 & \text{else.} \end{cases} \tag{3.54}
$$

!!!

As mentioned before we have a second possibility to derive the proximity operator for R_{MS}^*. For this we use Moreau's Theorem, c.f. Theorem **??**. It states, that we can compute $(\mathrm{Id} + \sigma\partial R_{MS}^*)^{-1}(\tilde{p})$ by

$$
(\mathrm{Id} + \sigma\partial R_{MS}^*)^{-1}(\tilde{p}) = \tilde{p} - \sigma(\mathrm{Id} + \frac{1}{\sigma}\partial R_{MS})^{-1}(\tilde{u}).
$$

Then we can compute the proximity operator of R_{MS} with

$$
(\mathrm{Id} + \tau\partial R_{MS})^{-1}(\tilde{u}) = \min_u \frac{||u-\tilde{u}||_2^2}{2} + \tau\min(\lambda||u||_2^2, \nu).
$$

First, we consider the case where the minimum function attains ν. Then

$$
\begin{aligned}
\min_u \frac{||u-\tilde{u}||_2^2}{2} + \nu\tau \quad &\Longleftrightarrow \quad \nabla\left(\frac{||u-\tilde{u}||_2^2}{2} + \nu\tau\right) = 0 \\
&\Longleftrightarrow \quad u - \tilde{u} = 0 \\
&\Longrightarrow \quad u = \tilde{u}.
\end{aligned}
$$

Now, assume that $\lambda||u||_2^2 < \nu$:

$$
\begin{aligned}
\min_u \frac{||u-\tilde{u}||_2^2}{2} + \lambda\tau||u||_2^2 \quad &\Longleftrightarrow \quad \nabla\left(\frac{||u-\tilde{u}||_2^2}{2} + \tau\lambda||u||_2^2\right) = 0 \\
&\Longleftrightarrow \quad (u - \tilde{u} + 2\lambda\tau u) = 0 \\
&\Longleftrightarrow \quad (1+2\lambda\tau)u = \tilde{u} \\
&\Longleftarrow \quad u = \frac{\tilde{u}}{(1+2\lambda\tau)}
\end{aligned}
$$

Again, since we know that at least $\lambda\frac{||u||_2^2}{2} \leq \nu$ holds we can compute the constraint by plugging $u = \frac{\tilde{u}}{(1+2\lambda\tau)}$ into the inequality. We observe

$$
\begin{aligned}
\lambda||u||_2^2 \leq \nu \iff & \lambda||\frac{\tilde{u}}{(1+2\lambda\tau)}||_2^2 \leq \nu \\
\iff & \frac{\lambda}{(1+2\lambda\tau)^2}||\tilde{u}||_2^2 \leq \nu \\
\iff & ||\tilde{u}||_2^2 \leq \frac{\nu}{\lambda}(1+2\lambda\tau)^2 \\
\implies & ||\tilde{u}||_2 \leq \sqrt{\frac{\nu}{\lambda}}(1+2\lambda\tau)
\end{aligned}
$$

The proximity operator for R_{MS} for all $i = 1, ..., N$ and $j = 1, ..., M$ is then given pointwise by

$$
u = (\text{Id} + \tau\partial R_{MS})^{-1}(\tilde{u}) \iff u_{i,j} = \begin{cases} \dfrac{1}{1+2\lambda\tau}\tilde{u}_{i,j} & \text{if } |\tilde{u}_{i,j}| \leq \sqrt{\frac{\nu}{\lambda}}(1+2\lambda\tau), \\ \tilde{u}_{i,j} & \text{else.} \end{cases} \tag{3.55}
$$

!! Computing the proximity operator for R_{MS}^* is now straightforward. We use Moreau's Theorem and get

$$
\begin{aligned}
(\text{Id} + \sigma\partial R_{MS}^*)^{-1}(\tilde{p}) &= \tilde{p} - \sigma\left(\text{Id} + \frac{1}{\sigma}\partial R_{MS}\right)^{-1}(\frac{\tilde{p}}{\sigma}) \\
&= \tilde{p} - \sigma\begin{cases} \dfrac{1}{1+\lambda\frac{1}{\sigma}}\dfrac{\tilde{p}}{\sigma} & \text{if } |\frac{\tilde{p}}{\sigma}| \leq \sqrt{\frac{\nu}{\lambda}}(1+\lambda\frac{1}{\sigma}), \\ \dfrac{\tilde{p}}{\sigma} & \text{else,} \end{cases} \\
&= \begin{cases} \tilde{p} - \dfrac{\sigma}{\sigma+\lambda}\tilde{p} & \text{if } \frac{1}{\sigma}|\tilde{p}| \leq \sqrt{\frac{\nu}{\lambda}}(1+\lambda\frac{1}{\sigma}), \\ 0 & \text{else,} \end{cases} \\
&= \begin{cases} \dfrac{\sigma+\lambda}{\sigma+\lambda}\tilde{p} - \dfrac{\sigma}{\sigma+\lambda}\tilde{p} & \text{if } |\tilde{p}| \leq \sqrt{\frac{\nu}{\lambda}}(\sigma+\lambda), \\ 0 & \text{else,} \end{cases} \\
&= \begin{cases} \dfrac{\lambda}{\sigma+\lambda}\tilde{p} & \text{if } |\tilde{p}| \leq \sqrt{\frac{\nu}{\lambda}}(\lambda+\sigma), \\ 0 & \text{else,} \end{cases}
\end{aligned}
$$

which is what we obtained before. Since the data fidelity term $G(u) = ||u-g||_2^2$ is uniformly convex we choose to solve this saddle-point problem with Algorithm **??**.

In this section we were able to provide a framework to solve highly non-convex Mumford-Shah functional. Unfortunately, there is no proof of convergence to the global minimum value. As we will see in Chapter 4 this lack of correctness does not matter. From a mathematical point of view, we still want to be able to compute the global

minimum of the Mumford-Shah functional. In the next chapter we will focus a framework, which again uses the primal-dual algorithm, that solves the functional optimally. !!!

4 A Primal-Dual Algorithm for Minimizing the Mumford-Shah Functional

As we revisit the Mumford-Shah Functional, we also rewrite Definition 2.1 to be consistend with [**?**], which is the publication we mainly follow in this chapter. If other results are taking into account, we make this clear.

Definition 4.1 (Mumford-Shah Functional) *Let $\Omega \subset \mathbb{R}^2$ be a rectangular image domain. In order to approximate an input image $f : \Omega \longrightarrow \mathbb{R}$ in terms of a piecewise smooth function $u : \Omega \longrightarrow \mathbb{R}$, Mumford and Shah suggested to minimize the functional*

$$E(u) = \lambda \int_{\Omega} (f-u)^2 dx + \int_{\Omega \setminus S_u} |\nabla u|^2 dx + \nu \mathcal{H}^1(S_u), \tag{4.1}$$

where $\lambda, \nu > 0$ are weighting parameters, $S_u = S_u^1 \cup ... \cup S_u^N$ and $\mathcal{H}^1(S_u)$ denotes the n-1-dimensional Hausdorff-measure of the curves in S_u.

The difference to Chapter 3 is, that we interchanged the set $K = K_1 \cup ... \cup K_N$ to $S_u = S_u^1 \cup ... \cup S_u^N$ and instead of computing $|K|$ the length of the curves we use the more general notation of measure theory. In the discrete case, the n-1-dimensional Hausdorff-measure of S_u is nothing but $|S_u|$. And the parameter which handles the tradeoff between the data fidelity term and the gradient based term is swapped. So both definitions are totally consistent. As mentioned in the previous chapter, this functional is highly non-convex. The idea of [**?**] was to use convex relaxation techniques to derive a convex formulation of the Mumford-Shah functional. Then we would again be able to apply our fast primal-dual algorithm.

4.1 Convex Relaxation

What we are looking for is the (global) minimal value of the piecewise smooth Mumford-Shah functional, i.e. $\min_{u \in X} E(u)$. Another approach would be to minimize the piecewise constant functional where we set the weight $\lambda = \infty$, which we also discussed in the last chapter. In order to find a convex relaxation we need to introduce the characteristic function.

Definition 4.2 *Let $\Omega \subset \mathbb{R}^2$ denote the image plane and let $u \in SBV(\Omega)$[1]. The upper level sets of u are denoted by the characteristic function $\mathbb{1}_u : \Omega \times \mathbb{R} \longrightarrow \{0,1\}$ of the subgraph of u:*

$$\mathbb{1}_u(x,t) = \begin{cases} 1, & \text{if } t < u(x), \\ 0, & \text{else.} \end{cases} \tag{4.2}$$

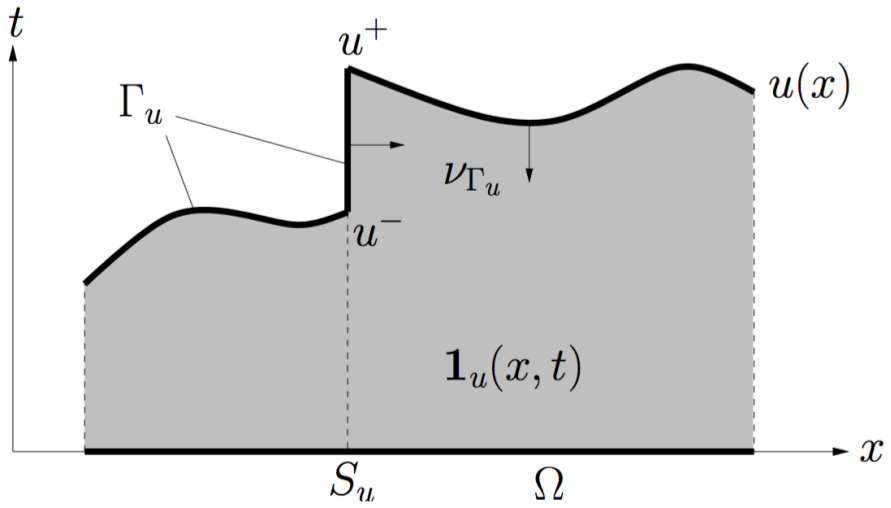


Figure 4.1: This picture (found in [**?**]) shows the characteristic function of another function $u(x)$. The gray shaded area shows where $\mathbb{1}_u(x,t) = 1$.

Remark 4.3 *This lifiting method together with the following theorem assures, that the Mumford-Shah Model we are facing will be convex. As soon as we have a convex optimazation problem we can apply the primal-dual algorithm to solve it.*

Using this characteristic function, one can find in a series of paper ([**?**], [**?**]) a result of the convex relaxed Mumford-Shah Functional along with a proof of this result.

Theorem 4.4 (Convex Relaxation of the Mumford-Shah Functional) *For a function $u \in SBV(\Omega)$ the Mumford Shah functional can be written as*

$$E(u) = \sup_{\varphi \in K} \int_{\Omega \times \mathbb{R}} \varphi D\mathbb{1}_u, \tag{4.3}$$

with a convex set

$$K = \Big\{ \varphi \in C^0(\Omega \times \mathbb{R}, \mathbb{R}^2) \quad : \quad \varphi^t(x,t) \geq \frac{\varphi^x(x,t)^2}{4} - \lambda(t - f(x))^2, \tag{4.4}$$

$$\left| \int_{t_1}^{t_2} \varphi^x(x,s) ds \right| \leq \nu \Big\}, \tag{4.5}$$

where the inequalities in the definition of K hold for all $x \in \Omega$ and for all $t, t_1, t_2 \in \mathbb{R}$.

Remark 4.5 *We used in the theorem, that we will represent a vector $\varphi \in K \subseteq \mathbb{R}^n$ by $\varphi(x,t) = \left(\varphi^x, \varphi^t\right)^T$, where $\varphi(x,t) \in \mathbb{R}^n$, $\varphi^x \in \mathbb{R}^{n-1}$ and $\varphi^t \in \mathbb{R}$.*

Our goal is now to minimize **??** since minimizing the energy of this functional should lead to an optimal approximation u of an input image f. Mathematically we have:

$$\min_{u\in X} E(u) = \min_{u\in X}\left(\sup_{\varphi\in K}\int_{\Omega\times\mathbb{R}}\varphi D\mathbb{1}_u\right) \tag{4.6}$$

To compute the minimizer of this formulation we first want to substitute $\mathbb{1}_u$ by a generic function

$$v(x,t):\Omega\times\mathbb{R}\longrightarrow[0,1] \text{ which satisfies } \lim_{t\to-\infty} v(x,t)=1,\quad \lim_{t\to+\infty} v(x,t)=0. \tag{4.7}$$

This substitution is important to take into account, that we assume an image u to be in $[0,1]$ in this chapter. Unfortunatelly, this makes the proposed method inexact. But still, we are computing a lower bound to the global optimal solution. Further, if we only consider the characteristic function then one can show that for binary images the lower bound is indeed the global optimal value of **??**. Overall, we are going to face the following, continous, convex optimization problem:

$$\min_{v\in[0,1]}\sup_{\varphi\in K}\langle v, D\varphi\rangle = \min_{v\in[0,1]}\sup_{\varphi\in K}\int_{\Omega\times\mathbb{R}}\varphi Dv. \tag{4.8}$$

In the next section we will reformulate this minimization problem in the discrete setting. Additionally, we will formulate this section in the sense of the saddle-point problem **??**.

4.2 Discrete Setting

Using the characteristic function, respectively the generic function v means, that we are adding an additional space to our two dimensional image domain. This extra label space needs to be considered in the discrete setting. In [**?**] they consider $\Omega=[0,1]^2$ and for that the subgraph of u to be in $[0,1]^3$. This would imply that we discetize these two spaces by adding a step-size h, where for instance $h=\frac{1}{N}$ or $h=\frac{1}{M}$. The size h would only scale the energy, but does not change results. Further, they consider all their operators without having this additional step-size. To be not confusing we propose all our spaces and operators without needing a step size. Then our image domain is $\Omega=\{1,2,...,N\}\times\{1,2,...,M\}$ and the subgraph of the function $u:\mathbb{R}^2\longrightarrow[0,1]$ is then defined in the cube $\Omega\times\{1,2,...,S\}$. In this discrete setting we define the pixel grid $\mathcal{G}$ with size $N\times M\times S$ and the following notation

$$\mathcal{G}=\left\{(i,j,k): i=1,2,...,N, j=1,...,M, k=1,2,...,S\right\} \tag{4.9}$$

where i,j,k are the discrete locations of each voxel. For a reformulation of **??** we also need to define the corresponding functions of v,φ. So, let $u\in X:\mathcal{G}\longrightarrow\mathbb{R}$ and $p\in Y:\mathcal{G}\longrightarrow\mathbb{R}^3$ be the discrete versions of the continous functions in equation **??** where u corresponds to v and p to φ. If we replace the inner-product for infinite dimensions in

equation **??** by the inner-product for finite dimensions and note that in finite spaces we can interchange sup and max we are going to face the saddle-point problem

$$\min_{u \in C} \max_{p \in K} \langle Au, p \rangle. \tag{4.10}$$

This notation looks now familiar to us. Here, A is our discrete, linear operator. We used the notation A instead of K as before, since we also have a discrete convex set K. Before we state how this set looks like, let us first give the definition of the set C.

$$C = \{u \in X : u(i,j,k) \in [0,1], u(i,j,1) = 1, u(i,j,S) = 0\} \subseteq X. \tag{4.11}$$

In equation **??** we substituted the characteristic function by a generic function v. To take the limits of v into account, we set the values in the first label space to 1 and those in the last label space to 0. We also stated, that our image u maps into $[0,1]$, which is also modeled in the set C. The discrete version of our set K from equation **??** can then be rewritten to

$$K = \{p = (p^x, p^t)^T \in Y \quad : \quad p^t(i,j,k) \geq \frac{||p^x||_2^2}{4} - \lambda(\frac{k}{M} - f(i,j))^2, \tag{4.12}$$

$$\left| \sum_{k_1 \leq k \leq k_2} p^x \right| \leq \nu\}, \tag{4.13}$$

whereas we define $p^x := (p^1, p^2)^T$ and $p^t := p^3$ and the vector p itself is an element of $\mathbb{R}^{N \times M \times S \times 3}$. For this, it also holds that $p^x \in \mathbb{R}^{N \times M \times S \times 2}$ and $p^t \in \mathbb{R}^{N \times M \times S}$. To be clear, the constraint in equation **??** goes pointwise for all $(i,j,k) \in \mathcal{G}$. The second constraint is more involved, since the constraint in **??** holds for all $i = 1, ..., N$, $j = 1, ..., M$ and all possible combinations (k_1, k_2) for all $k = 1, ..., S$. What looks like having a set K with two constraints, turns out that the set K is an intersection of a couple of convex sets. Namely, one has that for a fixed voxel (i,j,k) one can compute $\frac{S(S-1)}{2} + 1$ many convex sets. The amount of several convex sets will lead us to a long run-time for our algorithms, but also to interesting mathematical problems. Before we discuss this in detail, we first continuou by defining our discrete setting.

Remark 4.6 *In addition to discretize the variable t in* **??** *one gets* $\frac{k}{M}$ *in the discrete version for* **??***. Note, that t is a value in the continuous setting which determines at which point the characteristic function vanishes, i.e. is set to zero. The bound on the norm L depends on the discrete gradient operator and is for a fixed notation for this operator constant. Because of this, the notation one finds in [?] is wrong.*

We need to determine how the linear operator A looks like. It is the same as found in section **??**, namely the discrete gradient operator, but extended for the label space we added to this problem.

Definition 4.7 (Discrete gradient operator) *We define the discrete gradient of* $u \in X$ *by* $\nabla u = ((\partial_i u)_{i,j}, (\partial_j u)_{i,j}, (\partial_k u)_{i,j})^T$ *using forward differences with Neumann boundary conditions, i.e*

$$(\partial_i u)_{i,j} = \begin{cases} u_{i+1,j,k} - u_{i,j,k} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases} \quad (\partial_j u)_{i,j,k} = \begin{cases} u_{i,j+1,k} - u_{i,j,k} & \text{if } j < M \\ 0 & \text{if } j = M \end{cases}$$

$$(\partial_k u)_{i,j,k} = \begin{cases} u_{i,j,k+1} - u_{i,j,k} & \text{if } k < S \\ 0 & \text{if } k = S \end{cases}$$

Again we have $\text{div} : Y \longrightarrow X$ as the discrete divergence operator. And it relates to ∇ with $-\nabla^* = \text{div}$ as seen before.

Definition 4.8 (Discrete divergence operator) *We define the discrete divergence of* $p \in Y$ *by* $\nabla^T p = \partial_i p^1_{i,j,k} + \partial_j p^2_{i,j,k} + \partial_k p^3_{i,j,k}$ *using backward differences with Dirichlet boundary conditions, i.e*

$$(\partial_i p^1)_{i,j,k} = \begin{cases} p^1_{i,j,k} - p^1_{i-1,j,k} & \text{if } 1 < i < N \\ p^1_{i,j,k} & \text{if } i = 1 \\ -p^1_{i-1,j,k} & \text{if } i = N \end{cases} \quad (\partial_j p^2)_{i,j,k} = \begin{cases} p^2_{i,j,k} - p^2_{i,j-1,k} & \text{if } 1 < j < M \\ p^2_{i,j,k} & \text{if } j = 1 \\ -p^2_{i,j-1,k} & \text{if } j = M \end{cases}$$

$$(\partial_k p^3)_{i,j,k} = \begin{cases} p^3_{i,j,k} - p^3_{i,j,k-1} & \text{if } 1 < k < S \\ p^3_{i,j,k} & \text{if } k = 1 \\ -p^3_{i,j,k-1} & \text{if } k = S \end{cases}$$

Proposition 4.9 (Bound on the norm of ∇) *The bound on the norm of the proposed discrete linear operators is given by*

$$L^2 = ||\nabla|| = ||\text{div}|| \leq 12. \tag{4.14}$$

Proof The proof is the same as in section **??** by adding the additional discretization variable $p^3_{i,jk}$. ■

4.3 The Mumford-Shah Functional

4.3.1 The Mumford-Shah Model as Saddle-Point Problem

We already saw the formulation of the saddle-point problem in the section before. Recalling equation **??** and setting $A = \nabla$ we have that a minimizer of the Mumford-Shah Functional can be computed by solving

$$\min_{u \in C} \max_{p \in K} \langle \nabla u, p \rangle. \tag{4.15}$$

As in the chapter before we first want to formulate this saddle-point problem in the primal and the dual version. Further, we want to bring this primal-dual version into a

setting, where we can make use of one of our algorithms. At the end, we are able to compute the proximity operators and for that we give a setting to be able to implement the algorithm.

We can write equation **??** in a slightly different way, since we can simply add the indicator functions of the corresponding sets C and K. We observe

$$\min_{u\in C}\max_{p\in K}\langle\nabla u, p\rangle - \delta_K(p) + \delta_C(u). \tag{4.16}$$

But then, we immediately can determine our functions F^* and G, which are given by

$$F^*(p) = \delta_K(p) \text{ and } G(u) = \delta_C(u).$$

With this, we first want to take a look at the dual formulation of the Mumford-Shah model. The dual of the saddle-point problem was given by

$$\max_{p\in Y} -(G^*(-K^*p) + F^*(p)).$$

We need to compute the Legendre-Fenchel conjugate of G. Since this is the indicator function of the set C we can make use of Example **??** 1. and see that $G^*(p) = \delta_C^*(p) = \sup_{u\in C}\langle p, u\rangle$, which is the support function of C. Overall, we obtain the dual Mumford-Shah problem by

$$\max_{p\in Y} -(G^*(-K^*p) + F^*(p)) = \max_{p\in K\subset Y} -(\langle\nabla^T p, u\rangle + \delta_K(p)) = \max_{p\in K} -\langle\nabla^T p, u\rangle. \tag{4.17}$$

Finally, we also want to evaluate the primal formulation of the Mumford-Shah saddle-point problem. Therefore, it is left to compute F from F^*. Assume we have a function $F(\nabla u)$, then by definition the Legendre-Fenchel conjugate is given by

$$F^*(p) = \sup_{u\in C}\langle\nabla u, p\rangle - F(\nabla u).$$

On the other hand, if we assume F to be convex we observe

$$F(\nabla u) = \sup_{p\in K}\langle\nabla u, p\rangle - F^*(p) = \sup_{p\in K}\langle\nabla u, p\rangle - \delta_K(p)$$

which is exactly what we have in equation **??**. So, we want to compute $F(\nabla u)$ and show that the function F is indeed convex. Computing F means solving

$$\sup_{p\in K}\langle\nabla u, p\rangle - \delta_K(p) = \sup_{p\in K}\langle\nabla u, p\rangle.$$

We drop the indicator function, because in the case $p \notin K$ the function would be $\delta_K(p) = \infty$, and for that we would not attain the supremum. Also note that we are seeking for the supremum for all $p \in K$, for that the indicator function is zero. Back to our function F, we see that since K is finite dimensional, we can change the supremum to the maximum. We observe

$$\max_{p\in K}\langle\nabla u, p\rangle \iff \nabla\big(\nabla u, p\big) = 0 \iff \nabla u = 0.$$

But this means nothing that if the gradient of u vanishes, $F(\nabla u) = 0$. If the gradient is less of greater than zero the only choice for F over all p can only be $F(\nabla u) = \infty$. This holds of course for an arbitray argument of F, so that we have for a $u \in C$

$$F(u) = \begin{cases} 0 & \text{if } u = 0, \\ \infty & \text{else.} \end{cases}$$

This is nothing but the indicator function of a single point. Then for the primal formulation we get

$$\begin{aligned} \min_{u \in C} F(\nabla u) + G(u) \quad &= \quad \min_{u \in C} G(u) = 0 && (4.18) \\ &\text{s.t.} \quad \nabla u = 0. && (4.19) \end{aligned}$$

To solve the primal formulation one only needs to solve a linear equation, namely $\nabla u = 0$, meaning, that the minimum can only be attained if $F(\nabla u) = 0$ and for that $G(u) = 0$, otherwise G would be ∞.

4.3.2 The Proximity Operators for the Mumford-Shah Model

In this subsection we will see, that for the Mumford-Shah model we need to compute projections onto convex sets. This will then be discussed in the next section. But first let us note again that $F^*(p) = \delta_K(p)$ and $G(u) = \delta_C(u)$. From example **??** we know that the proximity operator of the indicator function is a projection onto the corresponding convex set. This implies, that the proximity operators for F^* and G are projections onto K and C, respectively. For that we want to rewrite our primal-dual algorithm to be consistent with [**?**]. We observe

Algorithm 4.10 *Choose $(u^0, p^0) \in C \times K$ and let $\bar{u}^0 = u^0$. We choose $\tau, \sigma > 0$. Then, we let for each $n \geq 0$*

$$\begin{cases} p^{n+1} = \Pi_K(p^n + \sigma K \bar{u}^n) \\ u^{n+1} = \Pi_C(u^n - \tau K^* p^{n+1}) \\ \bar{u}^{n+1} = 2u^{n+1} - u^n. \end{cases} \tag{4.20}$$

Here we denote Π_K and Π_C as the (euclidean) projections on the sets K and C.

4.4 Projection onto the convex sets and Dykstra's projection algorithm

In this section we want to discuss the projections onto the convex sets C and K. To project onto K, we additionally need Dykstra's project algorithm which we also discuss. We start with the projection onto the set C.

4.4.1 Projection onto C

The projection onto C can efficiently be computed. By definition of the proximity operator we have

$$u = \arg\min_{u \in C} \frac{||u - \tilde{u}||_2^2}{2} + \tau \delta_C(u) = \arg\min_{u \in C} \frac{||u - \tilde{u}||_2^2}{2}.$$

Assume that $\tilde{u} \in C$. Then the best choice for u is of course $u = \tilde{u}$ itself, since the energy of the minimization problem is equal to zero in this case, for which it is minimal. On the other hand, if $\tilde{u} \notin C$ the euclidean distance or shortest distance to $C = [0, 1]$ is to clip $\tilde{u}$ onto the bound of C. This means if $\tilde{u} < 0$ then the shortest distance from $\tilde{u}$ to u is to set $u = 0$. Reversely, if $\tilde{u} > 1$ then the euclidean distance to C is given by setting $u = 1$. This idea is also illustrated in figure (Bild einfuegen!!!). Overall, the projection onto C, which is also called clipping or l_∞-Projection is given by:

Algorithm 4.11 (Clipping) *The projection of a vector $u \in \mathbb{R}^{N \times M \times S}$ on the set*

$$C = \{u \in X : u(i, j, k) \in [0, 1],\ u(i, j, 1) = 1,\ u(i, j, M) = 0\}, \tag{4.21}$$

is given pointwise by

$$u_{i,j,k}^{n+1} = \min\{1, \max\{0, u_{i,j,k}^n\}\} \tag{4.22}$$

for all $i = 1, ..., N, j = 1, ..., M, k = 1, ..., S$.

Remark 4.12 *By projecting onto C we also need to take care about the limits in* **??**. *For that we set $u(i, j, 1) = 1$ and $u(i, j, S) = 0$ in each projection. Or similarly, using for instance the programming language C++, we would have $u(i, j, 0) = 1$ and $u(i, j, M - 1) = 0$.*

4.4.2 The projection onto K

Projecting onto K is slightly more difficult than the projection onto C since K takes into account local and non-local constraints. In other words, the set K is an intersection of several (convex) sets. The projection onto the intersection of convex sets can be done by Dykstra's projection algorithm. The idea behind this algorithm is to project onto each set n times alternatingly and store the error, which is made in each step. Before the projection in each step is done, the vector which should be projected is reduced by the error made in the previous step. To fully understand this scheme, we first give the definition which was first proposed by Boyle and Dykstra in [**?**], where one can also find a proof of convergence. Afterwards, we will provide and discuss the algorithm and additionally show an example.

Consider P convex sets with $\mathbb{R}^n \ni X = X_1 \cap X_2 \cap ... \cap X_P$. Let Π_i denote the projection onto the i-th set for $i = 1, ..., P$. And let $u_c \in \mathbb{R}^n$ be the current estimate with $u_c \notin X$, $u_i^k \in \mathbb{R}^n$ for $i = 0, ..., P$ and $v_i^k \in \mathbb{R}^n$ for $i = 1, ..., P$ and $k = 1, 2, ...$, where k denotes the number of iterations. Then the algorithm of (Boyle and) Dykstra finds for a point $u_c \in \mathbb{R}^n$ the (only) $u^* \in X$ such that

$$||u^* - u_c||^2 \leq ||u - u_c||^2 \ \forall u \in X.$$

Algorithm 4.13 *For $k = 1, 2, ...$ set $u_P^0 = u_c$ and $v_i^0 = 0$ for all $i = 1, ..., P$. Then iterate until convergence (e.g. $||u_0^k - u_P^k||_2 \leq \varepsilon$ with ε small):*

$$\begin{aligned} u_0^k = \ & u_P^{k-1}, \\ \text{for} \quad & i = 1, 2, ..., P : \\ & u_i^k = \Pi_i(u_{i-1}^k - v_i^{k-1}), \\ & v_i^k = u_i^k - (u_{i-1}^k - v_i^{k-1}). \end{aligned}$$

As mentioned the proof for the theorem of convergence can be found in [**?**]. The theorem is as follows:

Theorem 4.14 (Convergence) *The sequence u_0^k in algorithm **??** converges to the (only) point $u \in X$.*

(BILD!!!)

Recalling algorithm **??**, we see that in each iteration k we project the current vector u_{i-1}^k minus the error we made in the previous step, onto the i-th set. Afterwards, we update the error vector v_i^k. As illustrated in (BILD!!!), we project alternatingly on each set. At the end we convergence to the (only) point in K.

Now, that we know how the projection onto the entire set K can be implemented, we need to discuss how a projection of each of the several sets in K looks like. First let us give a decomposition of K into two several sets.

4.4.3 Decomposition of K

We will decompose the set K into two sets K_p and K_{nl}, where the first one resembles the local constraint, or more precisely, a parabola constraint and the second one corresponds to the non-local constraint. Overall, we have $K = K_p \cap K_{nl}$ and

$$K_p = \left\{ p^t(i,j,k) \geq \frac{||p^x(i,j,k)||^2}{4} - \lambda(\frac{k}{M} - f(i,j))^2 \right\} \ \forall i, j, k \tag{4.23}$$

where $p^t(i,j,k) = p^3(i,j,k)$ and $p^x(i,j,k) = (p^1(i,j,k), p^2(i,j,k))^T$ as mentioned before. For the non-local constraint we have

$$K_{nl} = \left\{ \left| \sum_{k_1 \leq k \leq k_2} p^x(i,j,k) \right| \leq \nu \right\} \ \forall i, j, k_1 \leq k \leq k_2 \tag{4.24}$$

We will now deduce the projection on these two sets. Let us start with the projection onto the parabola.

4.4.4 Projection onto K_p

Since the projection onto the set K_p is pointwise we want to drop the indices (i, j, k). Note that we do not necessarily need that a p^x is an element of $\mathbb{R}^2$. The following derivation holds for a larger class of problems namely having $p^x \in \mathbb{R}^n$. Let $\alpha > 0$, $p^x \in \mathbb{R}^n$, $p^t \in \mathbb{R}$ $p = (p^x, p^t)^T \in \mathbb{R}^n \times \mathbb{R}$. The projection of a p_0 with $p_0^t < \alpha ||p_0^x||_2^2$ can be written as

$$\begin{aligned} \min_{p \in \mathbb{R}^n \times \mathbb{R}} \quad & \frac{1}{2}||p - p_0||_2^2 \\ \text{subject to} \quad & p^t \geq \alpha ||p^x||_2^2 \end{aligned}$$

An equivalent notation for this problem is

$$\begin{aligned} \min_{p \in \mathbb{R}^n \times \mathbb{R}} \quad & f(p) = \frac{(p - p_0)^2}{2} \\ \text{subject to} \quad & g(p) = p^t - \alpha ||p^x||_2^2 \geq 0 \end{aligned}$$

To find the solution of this optimization problem we introduce some auxiliary variable (or Lagrange Multiplier) $\mu \in \mathbb{R}$ and define the Lagrangian as

$$\mathcal{L}(x, p, \mu) = f(p) - \mu g(p) = \frac{(p - p_0)^2}{2} - \mu \left(p^t - \alpha ||p^x||_2^2 \right). \tag{4.25}$$

Seeking for the local minima where $\nabla \mathcal{L}(x, p, \mu) = 0$ we even get a global minimum, since our function to optimize is convex, the inequality constraint is convex and the feasible set $\mathbb{R}^n \times \mathbb{R}$ is also convex. We set

$$\nabla \mathcal{L}(p, \mu) = \begin{pmatrix} \partial_{p^x} \mathcal{L}(p, \mu) \\ \partial_{p^t} \mathcal{L}(p, \mu) \\ \partial_{\mu} \mathcal{L}(p, \mu) \end{pmatrix} = \begin{pmatrix} p^x - p_0^x - \mu 2 \alpha p^x \\ p^t - p_0^t - \mu \\ p^t - \alpha ||p^x||_2^2 \end{pmatrix} = 0. \tag{4.26}$$

That means, we need to solve a linear system. The first equation gives us

$$p_0^x = (\mu 2\alpha + 1) p^x \iff p^x = \frac{p_0^x}{\mu 2\alpha + 1}, \tag{4.27}$$

and the second equation leads us to

$$p^t = p_0^t + \mu. \tag{4.28}$$

At this point we can solve this system in two ways - if $p_0^t \geq \alpha ||p_0^x||_2^2$ is not true already.

1. By plugging the two equalities **??** and **??** into the third line of equation **??**. We get

$$
\begin{aligned}
p_0^t + \mu - \alpha \left|\left| \frac{p_0^x}{\mu 2\alpha + 1} \right|\right|_2^2 = 0 \quad &\Longleftrightarrow \quad p_0^t + \mu - \frac{\alpha}{(\mu 2\alpha + 1)^2} ||p_0^x||_2^2 = 0 \\
&\Longleftrightarrow \quad (\mu 2\alpha + 1)^2 p_0^t + (\mu 2\alpha + 1)^2 \mu - \alpha ||p_0^x||_2^2 = 0 \\
&\Longleftrightarrow \quad (4\mu^2\alpha^2 + 4\mu\alpha + 1) p_0^t + 4\mu^3\alpha^2 + 4\mu^2\alpha + \mu - \alpha ||p_0^x||_2^2 = 0 \\
&\Longleftrightarrow \quad 4\alpha^2\mu^3 + \mu^2(4\alpha^2 p_0^t + 4\alpha) + \mu(4\alpha p_0^t + 1) + p_0^t - \alpha ||p_0^x||_2^2 = 0.
\end{aligned}
$$

Solving this equation is straightforward since computing the zeroes can be done by Newton's algorithm with

$$\mu^{k+1} = \mu^k - \frac{h(\mu)}{h'(\mu)}, \tag{4.29}$$

for $k = 1, 2, ...$

If we set

$$h(\mu) = 4\alpha^2\mu^3 + \mu^2(4\alpha^2 p_0^t + 4\alpha) + \mu(4\alpha p_0^t + 1) + p_0^t - \alpha ||p_0^x||_2^2,$$

we observe

$$h'(\mu) = 12\alpha^2\mu^2 + 2\mu(4\alpha^2 p_0^t + 4\alpha) + (4\alpha p_0^t + 1).$$

At the end we are having the update of a μ^{k+1} with

$$\mu^{k+1} = \mu^k - \frac{4\alpha^2\mu^3 + \mu^2(4\alpha^2 p_0^t + 4\alpha) + \mu(4\alpha p_0^t + 1) + p_0^t - \alpha ||p_0^x||_2^2}{12\alpha^2\mu^2 + 2\mu(4\alpha^2 p_0^t + 4\alpha) + (4\alpha p_0^t + 1)}. \tag{4.30}$$

In [**?**] they suggest setting $\mu^0 = \max\{0, -\frac{2p_0^t}{3}\}$, where they state that Newton's method then converges within 7-10 iterations to a quite accurate solution.
The projected vector p of our problem is then given by

$$p = \left(\frac{p_0^x}{\mu 2\alpha + 1}, p_0^t + \mu \right). \tag{4.31}$$

2. For the second approach we note that **??** and **??** hold and the third equation in **??** can only hold if

$$p^t = \alpha ||p^x||_2^2 \Longleftrightarrow \alpha ||p^x||_2^2 = p_0^t + \mu. \tag{4.32}$$

With **??** we can also compute the solution of μ by using that

$$
\begin{aligned}
||p^x||_2 = \left|\left| \frac{p_0^x}{1 + 2\alpha\mu} \right|\right|_2 \quad &\Longleftrightarrow \quad ||p^x||_2 = \frac{1}{1 + 2\alpha\mu} ||p_0^x||_2 \\
&\Longleftrightarrow \quad \frac{1}{1 + 2\alpha\mu} = \frac{||p^x||_2}{||p_0^x||_2} \\
&\Longleftrightarrow \quad 2\alpha\mu = \frac{||p_0^x||_2}{||p^x||_2} - 1 \\
&\Longleftrightarrow \quad \mu = \frac{1}{2\alpha} \left(\frac{||p_0^x||_2}{||p^x||_2} \right).
\end{aligned}
$$

Using the solution of μ in **??** we get

$$\begin{aligned}
\alpha||p^x||_2^2 = p_0^t + \frac{1}{2\alpha}\left(\frac{||p_0^x||_2}{||p^x||_2}\right) \quad &\overbrace{\iff}^{\cdot 2\alpha||p^x||_2} \quad 2\alpha^2||p^x||_2^3 = 2\alpha||p^x||_2 p_0^t + ||p_0^x||_2 - 1 \\
&\iff \quad 2\alpha^2||p^x||_2^3 + (1-2\alpha p_0^t)||p^x||_2 - ||p_0^x||_2 = 0. \\
&\overbrace{\iff}^{\cdot 4\alpha} \quad 8\alpha^3||p^x||_2^3 + 4\alpha(1-2\alpha p_0^t)||p^x||_2 - 4\alpha||p_0^x||_2 = 0. \\
&\iff \quad (2\alpha||p^x||_2)^3 + 2(1-2\alpha p_0^t)2\alpha||p^x||_2 - 4\alpha||p_0^x||_2 = 0. \\
&\iff \quad t^3 + 3bt - 2a = 0, \qquad (4.33)
\end{aligned}$$

with $a = 2\alpha||p_0^x||_2$, $b = \frac{2}{3}(1-2\alpha p_0^t)$ and $t = 2\alpha||p^x||_2$.
The cubic equation **??** in t can efficiently be solved using the analytical formulas for solving cubic equation published by J. P. McKelvey in 1984 in [**?**].

The result of the work mentioned is summarized in the following algorithm. Notice that we already computed the factors a and b. The others follow with [**?**].

Algorithm 4.15 *If already $p_0^t \geq \alpha||p_0^x||_2^2$, the solution is $(p^x, p^t) = (p_0^x, p_0^t)$. Otherwise, with $a = 2\alpha||p_0^x||_2$, $b = \frac{2}{3}(1-2\alpha p_0^t)$, and*

$$d = \begin{cases} a^2 + b^3 & \text{if } b \geq 0 \\ (a - \sqrt{-b}^3)(a + \sqrt{-b}^3) & \text{else} \end{cases}$$

set

$$v = \begin{cases} c - \dfrac{b}{c} \text{ with } c = \sqrt[3]{a + \sqrt{d}} & \text{if } d \geq 0 \\ 2\sqrt{-b}\cos\left(\dfrac{1}{3}\arccos\dfrac{a}{\sqrt{-b}^3}\right) & \text{else.} \end{cases}$$

If $c = 0$ in the first case, set $v = 0$. The solution is then given by

$$p^x = \begin{cases} \dfrac{v}{2\alpha}\dfrac{p_0^x}{||p_0^x||_2} & \text{if } p_0^x \neq 0 \\ 0 & \text{else} \end{cases}$$

and $p^t = \alpha||p^x||_2^2$.

The above method states that the projection onto the parabola can be done by one cycle of straightforward computations. The implementation of it is quite simple and computation time is fast, which we discuss in detail in chapter 4.

4.4.5 Projection onto K_{nl}

This set is a combination of non-local constraints, meaning that in a fixed point (i,j,k) you sum up for all $k_1 \leq k \leq k_2$. For that reason you can not project pointwise. First of all we want to present the algorithm:

Algorithm 4.16 (Soft Shrinkage Scheme) *Let $p_k^i = (p^1(i,j,k), p^2(i,j,k))^T \in \mathbb{R}^2$, $p^i = (p_1^i, ..., p_M^i)^T \in \mathbb{R}^{2\times M}$ for all $i = 1, 2, ...$ Then the projection p^{n+1} of a p^n - for an arbitrary, fixed pair (k_1, k_2) with $1 \leq k_1 \leq k \leq k_2 \leq M$ - is computed by:*

$$p^{n+1} = \begin{cases} p^n + \dfrac{s - \tilde{s}}{k_2 - k_1 + 1} & \text{if } k_1 \leq k \leq k_2, \\ p^n & \text{else,} \end{cases}$$

where $s \in \mathbb{R}^2, \tilde{s} \in \mathbb{R}^2$ with

$$\tilde{s} = \sum_{k_1 \leq k \leq k_2} p_k$$

and

$$s = \begin{cases} \tilde{s} & \text{if } ||\tilde{s}||_2 \leq \nu, \\ \Pi_{||\cdot||_2 \leq \nu}(\tilde{s}) = \dfrac{\nu}{||\tilde{s}||_2}\tilde{s} & \text{else.} \end{cases}$$

Since this procedure needs a clarification, we want to introduce the KKT conditions.

Theorem 4.17 (Karush-Kuhn-Tucker Optimality Conditions) *Consider the optimization problem*

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, i = 1, ..., m \end{aligned}$$

Defining the Lagrangian function to this optimization problem we have

$$\mathcal{L}(x, \lambda) = f(x) + \lambda, g(x), \tag{4.34}$$

with $\lambda \in \mathbb{R}^m$ are called the Lagrange-Multipliers.

Then if the objective function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ and the constraint functions $g_i : \mathbb{R} \longrightarrow \mathbb{R}$ are continously differentiable at a point x^* then it holds:

$$x^* \text{ is a local minimum} \iff \text{there exists a unique } \lambda^* \text{ such that}$$

- *Stationarity:*

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = 0$$

- *Complementary Slackness:*

$$\lambda_i^* g(x^*) = 0 \ \ \forall i = 1, ..., m$$

- *Primal Feasibility:*

$$g_i(x^*) \leq 0 \ \ \forall i = 1, ..., m$$

- *Dual Feasibility:*

$$\lambda_i^* \geq 0 \ \ \forall i = 1, ..., m$$

Proof of Algorithm ?? Let $p_k, \tilde{p}_k \in \mathbb{R}^2$ and $p, \tilde{p} \in \mathbb{R}^{2\times M}$ have the same form as in **??**. Then for a fixed pair (k_1, k_2) we face the following optimization problem:

$$\begin{aligned} \min_{p\in\mathbb{R}^{2\times M}} \quad & \frac{1}{2}||p - \tilde{p}||_2^2 \\ \text{subject to} \quad & \left|\left| \sum_{k_1\leq k\leq k_2} p_k \right|\right|_2 \leq \nu. \end{aligned}$$

This equivalent to

$$\begin{aligned} \min_{p\in\mathbb{R}^{2\times M}} \quad & f(p) \\ \text{subject to} \quad & g(p) \leq 0, \end{aligned} \tag{4.35}$$

if we set

$$f(p) = \sum_{k=1}^{M} \frac{(p_k - \tilde{p}_k)^2}{2}$$

and

$$g(p) = \frac{1}{2}\left|\left| \sum_{k_1\leq k\leq k_2} p_k \right|\right|_2^2 - \frac{1}{2}\nu^2.$$

This problem states that we try to find the closest p to $\tilde{p}$ whose components fullfil the inequality constraint in **??**.

Let us now introduce a new variable $\lambda \in \mathbb{R}$ which is called Lagrange Multiplier and observe the Lagrange function (or Lagrangian) with

$$\mathcal{L}(p, \lambda) = f(p) + \lambda g(p) \tag{4.36}$$

Looking at Stationarity condition to find a valid λ^* and p^* leads us to:

$$\nabla_p \mathcal{L}(p^*, \lambda^*) = \nabla_p f(p^*) + \lambda \nabla_p g(p^*) = \underbrace{\begin{pmatrix} p_1^* - \tilde{p}_1 \\ \vdots \\ p_M^* - \tilde{p}_M \end{pmatrix}}_{\in \mathbb{R}^M} + \lambda^* \underbrace{\begin{pmatrix} 0 \\ \vdots \\ 0 \\ \sum\limits_{k_1 \leq k \leq k_2} p_k^* \\ \vdots \\ \sum\limits_{k_1 \leq k \leq k_2} p_k^* \\ 0 \\ \vdots \\ 0 \end{pmatrix}}_{\in \mathbb{R}^M} = 0, \quad (4.37)$$

where the zeroes in the last vector are obtain for all components where $k < k_1$ and $k > k_2$. In these lines we already see that we get

$$p_k^* = \tilde{p}_k \text{ if } k < k_1 \text{ and } k > k_2.$$

Let us now take a closer look at the i-th line where $\nabla_{p_k} g(p) \neq 0$. Then we have

$$p_i^* - \tilde{p}_i + \lambda^* \sum_{k_1 \leq k \leq k_2} p_k^* = 0. \quad (4.38)$$

We set

$$\tilde{s} = \sum_{k_1 \leq k \leq k_2} \tilde{p}_k$$

and define a $s \in \mathbb{R}^2$ as

$$s = \begin{cases} \tilde{s} & \text{if } ||\tilde{s}||_2 \leq \nu, \\ \Pi_{||\cdot||_2 \leq \nu}(\tilde{s}) = \dfrac{\nu}{||\tilde{s}||_2} \tilde{s} & \text{else.} \end{cases}$$

For finding the optimal values p_k^* we want to distinguish between two cases:

1. $||\sum_{k_1 \leq k \leq k_2} \tilde{p}_k||_2 \leq \nu$:
   Setting $p_k^* = \tilde{p}_k + \frac{s-\tilde{s}}{k_2-k_1+1} = \tilde{p}_k$ for all $k_1 \leq k \leq k_2$ and $s = \tilde{s}$ leads to

$$\frac{1}{2}||\sum_{k_1 \leq k \leq k_2} p_k^*||_2^2 - \frac{1}{2}\nu^2 \leq 0, \tag{4.39}$$

   It follows for the Complementary Slackness condition that we can choose an arbitrary $\lambda^* \geq 0$ if we have equality in **??** and $\lambda^* = 0$ else to derive the Dual Feasibility condition. We set

$$\lambda^* = 0.$$

2. $||\sum_{k_1 \leq k \leq k_2} \tilde{p}_k||_2 > \nu$:
   Here, we would violate the Primal Feasibility condition if we set $p_k^* = \tilde{p}_k$ for all $k_1 \leq k \leq k_2$. For this reason we choose

$$p_k^* = \tilde{p}_k + \frac{s-\tilde{s}}{k_2-k_1+1} \quad \forall k_1 \leq k \leq k_2$$

   We observe for the Primal Feasibility condition that

$$\begin{aligned} \frac{1}{2}||\sum_{k_1 \leq k \leq k_2} p_k^*||_2^2 - \frac{1}{2}\nu^2 &= \\ &= \frac{1}{2(k_2-k_1+1)}||\sum_{k_1 \leq k \leq k_2} \tilde{p}_k + (s-\tilde{s})||_2^2 - \frac{1}{2}\nu^2 \\ &= \frac{1}{2}||\sum_{k_1 \leq k \leq k_2} \tilde{s} + s - \tilde{s}||_2^2 - \frac{1}{2}\nu^2 = \frac{1}{2}\underbrace{||s||_2^2}_{=\nu^2} - \frac{1}{2}\nu^2 = 0. \end{aligned} \tag{4.40}$$

   Using this equality we already see that Complementary Slackness is fullfild with a dual feasible $\lambda^* \geq 0$. Since we set $p_k^* = \tilde{p}_k + \frac{s-\tilde{s}}{k_2-k_1+1}$ we can plug p^* into **??** to derive λ^*. It follows

$$
\begin{aligned}
& p_i^* - \tilde{p}_i + \frac{\lambda^*}{k_2-k_1+1} \sum_{k_1 \le k \le k_2} p_k^* = \\
= \;& \tilde{p}_i + \frac{s-\tilde{s}}{k_2-k_1+1} - \tilde{p}_i + \frac{\lambda^*}{k_2-k_1+1} \sum_{k_1 \le k \le k_2} \tilde{p}_k + \frac{s-\tilde{s}}{k_2-k_1+1} \\
= \;& \frac{s-\tilde{s}}{k_2-k_1+1} + \frac{\lambda^*}{k_2-k_1+1} \Bigg(\underbrace{\sum_{k_1 \le k \le k_2} \tilde{p}_k}_{=\tilde{s}} + \underbrace{\sum_{k_1 \le k \le k_2} \frac{s-\tilde{s}}{k_2-k_1+1}}_{=\frac{(k_2-k_1+1)(s-\tilde{s})}{k_2-k_1+1}=s-\tilde{s}} \Bigg) \\
= \;& \frac{s-\tilde{s}}{k_2-k_1+1} + \frac{\lambda^*}{k_2-k_1+1}(\tilde{s}+s-\tilde{s}) \\
= \;& \frac{s-\tilde{s}}{k_2-k_1+1} + \frac{\lambda^*}{k_2-k_1+1} s \\
= \;& \frac{1}{k_2-k_1+1}(s-\tilde{s}+\lambda^* s) = 0 \qquad (4.41)
\end{aligned}
$$

Now we can solve for λ^* using that the last equation **??** is equivalent to

$$
\begin{aligned}
(s-\tilde{s}+\lambda^* s) = 0 \quad & \overbrace{\Longleftrightarrow}^{||\tilde{s}||_2 > \nu} \quad \left(\frac{\nu}{||\tilde{s}||_2}\tilde{s} - \tilde{s} + \lambda^* \frac{\nu}{||\tilde{s}||_2}\tilde{s} \right) = 0 \\
& \Longleftrightarrow \quad \tilde{s}\left(\frac{\nu}{||\tilde{s}||_2} - 1 + \lambda^* \frac{\nu}{||\tilde{s}||_2} \right) = 0 \\
& \Longleftrightarrow \quad \frac{\nu}{||\tilde{s}||_2} - 1 + \lambda^* \frac{\nu}{||\tilde{s}||_2} = 0 \\
& \Longleftrightarrow \quad \lambda^* \frac{\nu}{||\tilde{s}||_2} = 1 - \frac{\nu}{||\tilde{s}||_2} \\
& \Longleftrightarrow \quad \lambda^* = \underbrace{\frac{||\tilde{s}||_2}{\nu}}_{>1} - 1 > 0.
\end{aligned}
$$

This satisfies the Dual Feasibility condition.

With the choices of $\tilde{s}, s$ and

$$
p_k^* = \tilde{p}_k + \frac{s-\tilde{s}}{k_2-k_1+1} \quad \forall k_1 \le k \le k_2
$$

we solved **??** (Stationarity). Applying this procedure to each combination of the (k_1, k_2) and replacing p^* by p^{n+1}, $\tilde{p}$ by p^n respectively, we observe our algorithm. ■

4.4.6 An alternative approach using Lagrange Multiplier

Let $f(x,y) := \langle Ax, y\rangle$. Hence we have

$$\min_{x\in C}\max_{y\in K} f(x,y) = \min_{x\in C}\max_{y\in K}\langle Ax, y\rangle. \tag{4.42}$$

Rewrite non-local-constraint in the set K - **??** is pointwise so this is already easy to compute:
We get the formulation

$$||p_{k_1,k_2}||_2 \leq \nu \;\text{ s.t. }\; p_{k_1,k_2} = \sum_{k_1\leq k\leq k_2} (y_1(i,j,k), y_2(i,j,k))^T \tag{4.43}$$

which corresponds to **??**.
The constraint functions in **??** is equivalent to

$$g(p,y) := p_{k_1,k_2} - \sum_{k_1\leq k\leq k_2} (y_1(i,j,k), y_2(i,j,k))^T = 0 \tag{4.44}$$

holding for all combinations of k_1, k_2 with $1 \leq k_1 \leq k \leq k_2 \leq M$.
Let $K = \frac{M(M-1)}{2}$ be all possible combinations of the k_1, k_2. For a Lagrange (dual) function $\mathcal{L}$ depending on the optimization problem in **??** we get

$$\begin{aligned}\mathcal{L}(x,y,\lambda,p) &= f(x,y) + \sum_K \lambda_K g(p,y) = \\ &\langle Ax, y\rangle + \sum_{k_1=1}^{M}\sum_{k_2=k_1}^{M}\langle \lambda_{k_1,k_2}, p_{k_1,k_2} - \sum_{k_1\leq k\leq k_2} (y_1(i,j,k), y_2(i,j,k))^T\rangle\end{aligned} \tag{4.45}$$

where $x = x(i,j,k) \in \mathbb{R}^{N\times N\times M}, y(i,j,k) = (y_1(i,j,k), y_2(i,j,k), y_3(i,j,k))^T \in \mathbb{R}^{3\times N\times N\times M}, \lambda(i,j,k) \in \mathbb{R}^{2\times K\times N\times N}$ and $p(i,j,k) \in \mathbb{R}^{2\times K\times N\times N}$.
Now, instead of minimizing over x and maximizing over y we have an equivalent formulation:

$$\min_{x\in C}\max_{y\in K} f(x,y) \Longleftrightarrow \min_{\substack{x\in C\\ \lambda_{k_1,k_2}}}\max_{\substack{y\in K_p\\ ||p_{k_1,k_2}||\leq\nu}} \mathcal{L}(x,y,\lambda,p), \tag{4.46}$$

where K_p denotes the subset of K with only the parabola constraint, i.e.

$$\begin{aligned}K = \{&y = (y^1, y^2, y^3)^T \in Y : \\ &y^3(i,j,k) \geq \frac{y^1(i,j,k)^2 + y^2(i,j,k)^2}{4} - \lambda(\frac{k}{M} - f(i,j))^2.\end{aligned} \tag{4.47}$$

To solve this saddle-point problem we can again apply our primal dual algorithm. As mentioned we can solve problems of the form $\min\max\langle Ax, y\rangle$ with this algorithm.
Now, let $\tilde{x} = (x,\lambda)^T$ and $\tilde{y} = (y,p)^T$ then we have

Algorithm 4.18 *Choose* $(\tilde{x}^0, \tilde{y}^0) \in \tilde{C} \times \tilde{K}$ *and let* $\bar{\tilde{x}}^0 = \tilde{x}^0$. *We choose* $\tau, \sigma > 0$. *Then, we let for each* $n \geq 0$

$$\begin{cases} \tilde{y}^{n+1} = \Pi_{\tilde{K}}(\tilde{y}^n + \sigma A \bar{\tilde{x}}^n) \\ \tilde{x}^{n+1} = \Pi_{\tilde{C}}(\tilde{x}^n - \tau A^* \tilde{y}^{n+1}) \\ \bar{\tilde{x}}^{n+1} = 2\tilde{x}^{n+1} - \tilde{x}^n. \end{cases} \tag{4.48}$$

The question how to compute the new projections and to bring the λ and p into this algorithm remains. To give the answer to this question we first need to differentiate the Lagrange function, to get the derivative.

$$\frac{\partial \mathcal{L}(x,y,\lambda,p)}{\partial x} = A^T y \tag{4.49}$$

$$\frac{\partial \mathcal{L}(x,y,\lambda,p)}{\partial p} = \sum_{k_1,k_2} \lambda_{k_1,k_2} \tag{4.50}$$

$$\frac{\partial \mathcal{L}(x,y,\lambda,p)}{\partial \lambda} = \sum_{k_1,k_2} \left(p_{k_1,k_2} - \sum_{k_1 \leq k \leq k_2} (y_1(i,j,k), y_2(i,j,k))^T \right) \tag{4.51}$$

$$\frac{\partial \mathcal{L}(x,y,\lambda,p)}{\partial y} = Ax + \hat{y}. \tag{4.52}$$

To see what $\hat{y}$ looks like, we need to compute the partial derivative of $\mathcal{L}$ in the y-direction. By looking at the l-th component we observe:

$$\begin{aligned} \frac{\partial \mathcal{L}(x,y,\lambda,p)}{\partial y_l} &= (Ax)_l + \frac{\partial}{\partial y_l} \left(\sum_{k_1=1}^{M} \sum_{k_2=k_1}^{M} \langle \lambda_{k_1,k_2}, p_{k_1,k_2} - \sum_{k_1 \leq k \leq k_2} (y_1(i,j,k), y_2(i,j,k))^T \right) \\ &= (Ax)_l + \frac{\partial}{\partial y_l} \left(\sum_{k_1=1}^{M} \sum_{k_2=k_1}^{M} \langle \lambda_{k_1,k_2}, p_{k_1,k_2} \rangle - \langle \lambda_{k_1,k_2}, \sum_{k_1 \leq k \leq k_2} (y_1(i,j,k), y_2(i,j,k))^T \right) \\ &= (Ax)_l - \frac{\partial}{\partial y_l} \left(\sum_{k_1=1}^{M} \sum_{k_2=k_1}^{M} \langle \lambda_{k_1,k_2}, \sum_{k_1 \leq k \leq k_2} (y_1(i,j,k), y_2(i,j,k))^T \right) \\ &= (Ax)_l - (\lambda_l^1 + \lambda_l^2) \end{aligned} \tag{4.53}$$

We observe the following algorithm:

Algorithm 4.19 *Choose* $(x^0, y^0, \lambda^0, p^0) \in C \times K_p \times \mathbb{R}^{2 \times N \times N \times M} \times \mathbb{R}^{2 \times N \times N \times M}$ *and let* $\bar{x}^0 = x^0, \bar{\lambda}^0 = \lambda^0$. *We choose* $\tau_x = \frac{1}{6}, \tau_\lambda = \frac{1}{2+k_2-k_1}, \sigma_y = \frac{1}{3+M}, \sigma_p = 1$. *Then, we let for each* $n \geq 0$

$$\begin{cases} y^{n+1} = \Pi_{K_p}(y^n + \sigma_y(A\bar{x}^n + \tilde{y})) \\ p_{k_1,k_2}^{n+1} = \Pi_{||\cdot||_2 \leq \nu}(p_{k_1,k_2}^n + \sigma_p \bar{\lambda}_{k_1,k_2}^n) \\ x^{n+1} = \Pi_C(x^n - \tau_x A^* y^{n+1}) \\ \lambda_{k_1,k_2}^{n+1} = \lambda_{k_1,k_2}^n - \tau_\lambda(p_{k_1,k_2}^{n+1} - \sum_{k_1 \leq k \leq k_2} (y_1(i,j,k), y_2(i,j,k))^T) \\ \bar{x}^{n+1} = 2x^{n+1} - x^n \\ \bar{\lambda}_{k_1,k_2}^{n+1} = 2\lambda_{k_1,k_2}^{n+1} - \lambda_{k_1,k_2}^n. \end{cases} \tag{4.54}$$

5 Applications to Imaging

(a) $\lambda = 0.06$ (b) $\lambda = 0.07$

(c) $\lambda = 0.09$ (d) $\lambda = 0.1$

(e) $\lambda = 0.6$ (f) $\lambda = 0.9$

Figure 5.1: Approximation of Audrey Hepburn with the ROF Model. The higher λ the closer the outcome of the algrithm to the original image.

(a) $\lambda = 0.06$ (b) $\lambda = 0.07$

(c) $\lambda = 0.09$ (d) $\lambda = 0.1$

(e) $\lambda = 0.6$ (f) $\lambda = 0.9$

Figure 5.2: Approximation of the Lena image with the ROF Model. Again, the higher λ the closer the outcome of the algrithm to the original image.

(a) $\lambda = 0.6$ (b) $\lambda = 0.7$

(c) $\lambda = 0.8$ (d) $\lambda = 0.9$

Figure 5.3: Approximation of Audrey Hepburn with the TVL1 Model. The higher λ the closer the outcome of the algrithm to the original image.

(a) $\lambda = 0.6$ (b) $\lambda = 0.7$

(c) $\lambda = 0.8$ (d) $\lambda = 0.9$

Figure 5.4: Approximation of the Lena image with the TVL1 Model. Again, the higher λ the closer the outcome of the algrithm to the original image.

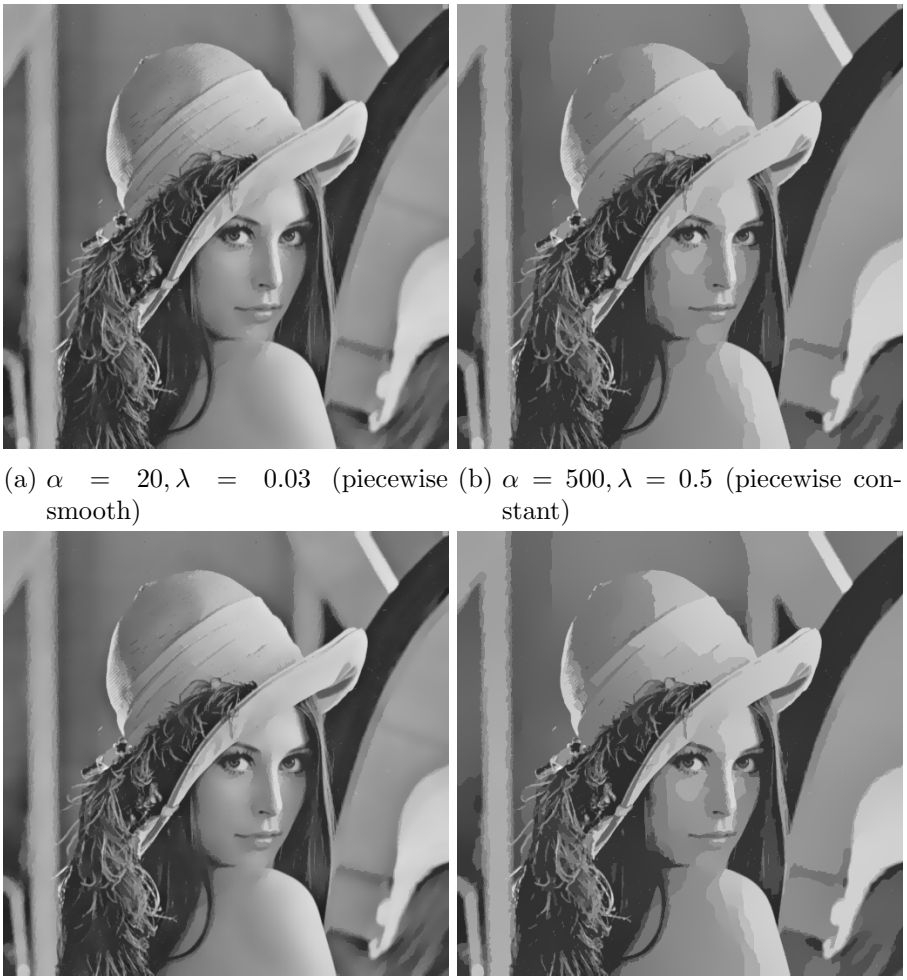
(a) $\alpha = 20, \lambda = 0.03$ (piecewise smooth)

(b) $\alpha = 500, \lambda = 0.5$ (piecewise constant)

(c) $\alpha = 20, \lambda = 0.04$ (piecewise smooth)

(d) $\alpha = 500, \lambda = 0.7$ (piecewise constant)

Figure 5.5: Approximation of Audrey Hepburn with the Mumford-Shah Model. The higher λ the closer the outcome of the algrithm to the original image.



(a) $\alpha = 20, \lambda = 0.03$ (piecewise smooth)

(b) $\alpha = 500, \lambda = 0.5$ (piecewise constant)

(c) $\alpha = 20, \lambda = 0.04$ (piecewise smooth)

(d) $\alpha = 500, \lambda = 0.7$ (piecewise constant)

Figure 5.6: Approximation of the Lena image with the Mumford-Shah Model. Again, the higher λ the closer the outcome of the algrithm to the original image.

Figure 5.7: Evolution of the Lena image, whereas the left column shows the ROF Model, the second column the TVL1 Model, the third column resembles the piecewise smooth approximation of the Mumford-Shah Model with the real-time minimizer with $\alpha = 20$. The right column shows the piecewise constant case of the last mentioned algorithm with $\alpha = 500$.

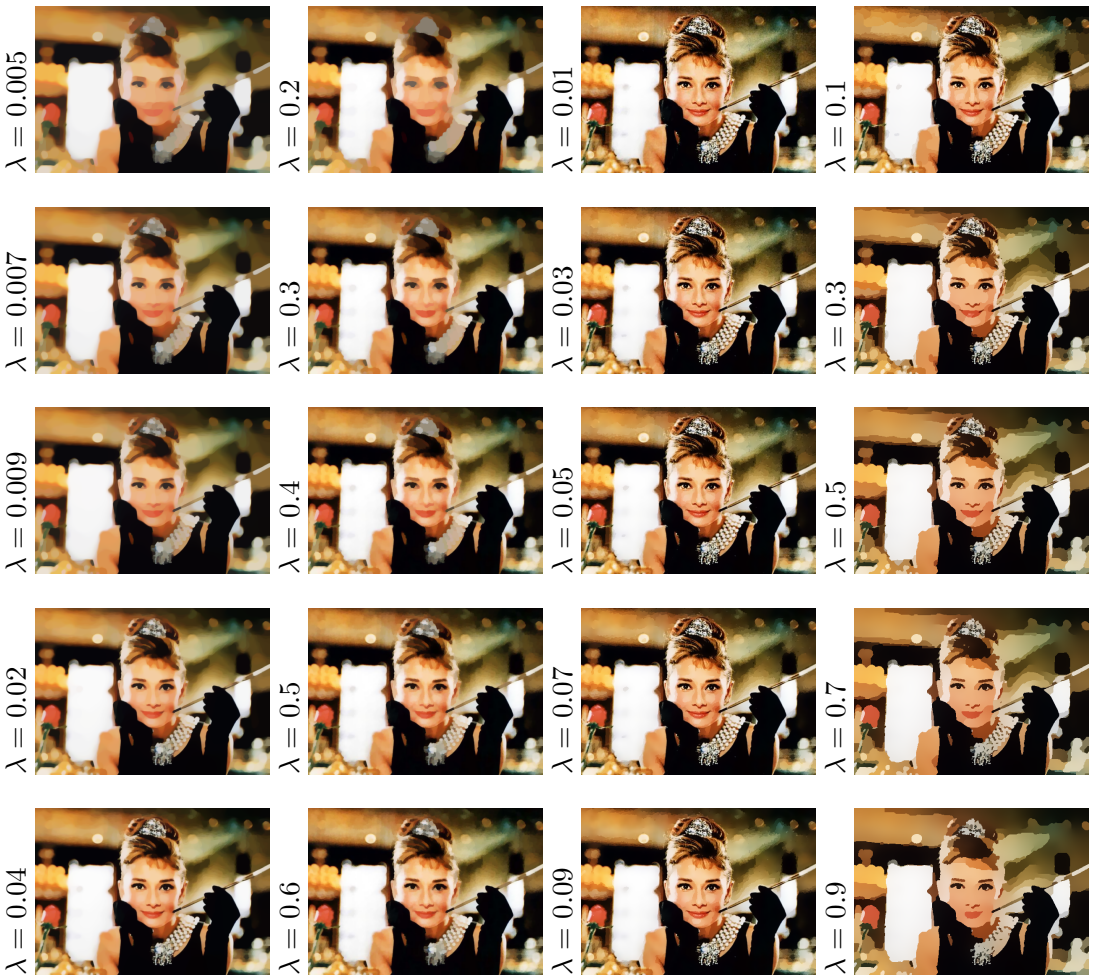


Figure 5.8: Evolution of the Audrey Hepburn image, whereas the left column shows the ROF Model, the second column the TVL1 Model, the third column resembles the piecewise smooth approximation of the Mumford-Shah Model with the real-time minimizer with $\alpha = 20$. The right column shows the piecewise constant case of the last mentioned algorithm with $\alpha = 500$.

(a) $\alpha = 500, \lambda = 0.4, \gamma = 1$ (b) $\alpha = 500, \lambda = 0.4, \gamma = 10$

(c) $\alpha = 500, \lambda = 0.4, \gamma = 20$ (d) $\alpha = 500, \lambda = 0.5, \gamma = 50$

Figure 5.9: Cartooning of Audrey Hepburn with the Mumford-Shah Model. The higher λ the closer the outcome of the algrithm to the original image.

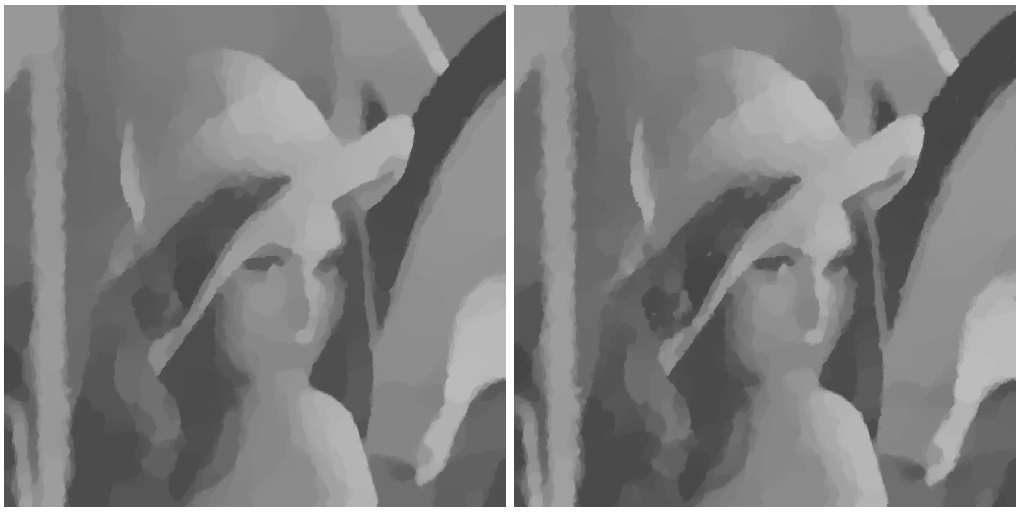
(a) $\lambda = 0.02$ (b) $\lambda = 0.03$

Figure 5.10: Denoising of Salt and Pepper noise of the Lena image with the ROF Model.

(a) $\lambda = 0.5$ (b) $\lambda = 0.6$

(c) $\lambda = 0.7$ (d) $\lambda = 0.8$

Figure 5.11: Denoising of Gaussian noise of the Lena image with the TVL1 Model.



(a) $\lambda = 0.007$ (b) $\lambda = 0.008$

Figure 5.12: Denoising of Salt and Pepper noise of the Lena image with the ROF Model.

(a) $\lambda = 0.5$ (b) $\lambda = 0.6$

(c) $\lambda = 0.7$ (d) $\lambda = 0.8$

Figure 5.13: Denoising of Salt and Pepper noise of the Lena image with the TVL1 Model.

(a) $\lambda = 0.06$ (b) $\lambda = 0.07$

(c) $\lambda = 0.09$ (d) $\lambda = 0.1$

Figure 5.14: Inpainting of Lena image with the ROF Model.

Bibliography

[1] E. Strekalovskiy and D. Cremers. Real-time minimization of the piecewise smooth mumford-shah functional. In *European Conference on Computer Vision (ECCV)*, pages 127–141, 2014. ¡a href=”https://github.com/tum-vision/fastms”¿Code available¡/a¿.