Westfälische
Wilhelms-Universität
Münster

Discretization via line measures for a curvature regularization framework

Master Thesis

submitted in fulfillment of the requirements for the degree of

Master of Science

University of Münster, Germany
Department of Mathematics and Computer Sciences
Institute for Computational and Applied Mathematics

Supervisor and first assessor:
Prof. Dr. Benedikt Wirth
Second assessor:
Prof. Dr. Martin Burger

Submitted by:
Daniel Tinius

Münster, September 2015

Abstract

In this thesis we propose a new discretization scheme for an existing convex curvature regularization framework for image processing. The essential idea is to apply regularity on the level lines of images, which is accomplished using functional lifting of the image gradient into a Radon measure in three-dimensional space $\Omega \times S^1$, where Ω is the image domain, and the normals to the level lines are represented in S^1. Utilizing the special structure of this higher-dimensional measure we propose a new discretization based on linear combinations of short line measures, and present two efficient methods for numerically solving the resulting linear optimization problems.

Through various examples including deconvolution, binary segmentation, inpainting and a specialized noise removal we show the diversity of the framework and the effectiveness of the new discretization.

Declaration of Academic Integrity

I hereby confirm that this thesis on *Discretization via line measures for a curvature regularization framework* is solely my own work and that I have used no sources or aids other than the ones stated. All passages in my thesis for which other sources, including electronic media, have been used, be it direct quotes or content references, have been acknowledged as such and the sources cited.

(date and signature of student)

I agree to have my thesis checked in order to rule out potential similarities with other works and to have my thesis stored in a database for this purpose.

(date and signature of student)

Contents

1 Introduction **1**

2 Prerequisites **2**

2.1 Measure theory 2

2.1.1 Radon-measures and functions of bounded variation 2

2.1.2 Distance functions on measures 3

2.2 Convex optimization 6

3 Image Regularization Framework **10**

3.1 Functional lifting 10

3.2 Functionals acting on polygons 12

3.3 Generalization to admissible polygons 15

3.3.1 Generalization to imaging problems 18

3.4 Convex Relaxation 19

3.5 Functionals TVX_0 and TVX_1 20

3.6 Applications for imaging problems with L^2 data term 20

3.7 Data terms 21

3.7.1 Noise removal 21

3.7.2 Binary segmentation 21

3.7.3 Inpainting 22

3.7.4 Deconvolution 22

3.7.5 Thin structures preserving noise removal 22

4 Discretization via Line Measures **26**

4.1 Discrete setting 26

4.1.1 Discrete compatibility condition 28

4.1.2 Discrete directional derivative 30

4.1.3 Discrete partial derivative in label direction 31

4.1.4 Discrete flow balancing condition 32

4.1.5 Discrete projection 32

4.2 Matrix representation of discrete operators 33

4.3 Discrete formulations for TVX 35

4.3.1 Discrete TVX_0 35

4.3.2 Discrete TVX_1 36

5 Numerical algorithms **38**

5.1 Primal-Dual algorithm 38

5.1.1 TVX0-PD 40

5.1.2 TVX1-PD 41

5.2 Linear programming 42

5.2.1 Generalization for quadratic data terms 42

5.2.2 Transformation of norm minimization problems 43

5.2.3 TVX0-LP 45

5.2.4 TVX1-LP 46

6 Applications and Evaluation 47

6.1 Numerical results 47

6.1.1 Noise removal 47

6.1.2 Binary segmentation 49

6.1.3 Inpainting 51

6.1.4 Deconvolution 53

6.1.5 Thin structures preserving noise removal 54

6.2 Comparison of discretizations 58

6.3 Performance evaluations 60

6.3.1 Implementation details 60

6.3.2 Memory consumption 60

6.3.3 Running times 62

6.3.4 Performance conclusions 65

7 Conclusions 67

List of Figures 69

References 71

1 Introduction

Many applications in image processing rely on additional information introduced through regularization. While for a long time first-order smoothness conditions have been predominant, more recent research has shown that higher-order features such as curvature are often better suited for solving problems such as image segmentation, inpainting or noise removal. The underlying assumption is that level lines of images are mostly smooth with only small curvature.

This thesis discusses one approach to curvature regularization that has been proposed in [7]. The essential idea is to impose regularity on the level lines of images by assigning penalties to all vertices where two level lines meet. The penalty is based on a metric distance between the incoming and outgoing orientations of the two connected level lines. Hence the authors considered functionals that each depend on one particular metric in S^1 and are defined as the sum of all vertex penalties. The article proposes two exemplary penalty functionals where the first one counts all vertices, and the second one measures the total sum of the absolute external angles, which in case of smooth level lines corresponds to the total absolute curvature.

The main focus of this work is a new numerical discretization for the curvature regularization framework, that was designed to capture the special structure of these functionals. To obtain a convex representation for the vertex penalizing functionals the authors used a technique called 'functional lifting' to represent image gradients by higher-dimensional Radon measures. These measures are essentially composed of line measures, an observation that will prove crucial to the proposed method. We will approximate the lifted measures using linear combinations of short, local line measures and show that this approach can be solved with high efficiency using different numerical algorithms including standard linear programming.

We will proceed as follows. The second chapter will introduce the main mathematical concepts including an introduction to Radon measure theory and convex analysis. The third chapter will then summarize the derivation of the curvature regularization framework and propose a few sample applications. In the fourth and fifth chapter we will introduce the new discretization and show different formulations that can be solved using different algorithms and solvers. The last chapter will then contain numerical results for all of the proposed example applications, and an evaluation that compares the new method to the original one. Moreover we study two different methods of solving the example problems in terms of practical usability and performance comparisons.

2 Prerequisites

In this first chapter we will summarize the main mathematical concepts that will be used throughout this thesis. The first part covers a condensed introduction into Radon-measure theory and functions of bounded variations, which will be substantial for the derivation of the curvature regularization framework. The second part will focus on convex analysis and especially convex duality theory that will be helpful for the numerical discretization.

2.1 Measure theory

2.1.1 Radon-measures and functions of bounded variation

In this thesis we will consider a regularization framework based on measures and therefore we will start by introducing a few common definitions.

In the following let $\Omega \subset \mathbb{R}^2$ be a connected open set, i.e. a domain. With S^1 we denote the unit sphere in $\mathbb{R}^2$, $C_c(X)$ the space of continuous functions with compact support and $C_0(X)$ its completion with respect to the sup-norm. Further we denote by $\mathcal{B}(\Omega)$ the Borel algebra generated by the open subsets of Ω.

Definition 2.1 (Total Variation)**.** *The* ***total variation*** *of a function $u \in L^1(\Omega)$ is defined as*

$$\mathrm{TV}(u) = \sup \left\{ \int_\Omega u \operatorname{div} \varphi \, \mathrm{d}x \;\middle|\; \varphi \in C_c^\infty(\Omega, \mathbb{R}^2), \|\varphi\|_\infty \leq 1 \right\}. \tag{2.1}$$

The function u is said to be of ***bounded variation*** *if its total variation is finite, i.e.* $\mathrm{TV}(u) < \infty$.

Definition 2.2 (Space BV)**.** *The* ***space of functions of bounded variation*** *is the set*

$$\mathrm{BV}(\Omega) = \left\{ u \in L^1(\Omega) \;\middle|\; \mathrm{TV}(u) < \infty \right\}.$$

Together with the norm $\|u\|_{\mathrm{BV}} = \|u\|_1 + \mathrm{TV}(u)$ it is a Banach space.

Definition 2.3 (Finite Radon Measures)**.** *A set function $\mu : \mathcal{B}(\Omega) \to \mathbb{R}^d$ is called a* ***finite Radon measure*** *if it is countably additive, regular and satisfies $\mu(\emptyset) = 0$.*

A finite Radon measure $\mu : \mathcal{B}(\Omega) \to \mathbb{R}$ is called ***positive****, denoted by $\mu \geq 0$, if $\mu(E) \geq 0$ for all $E \in \mathcal{B}(\Omega)$.*

Definition 2.4 (Total variation measure)**.** *Let $\mu : \mathcal{B}(\Omega) \to \mathbb{R}^d$ be a finite Radon measure. Then the* ***total variation measure*** *$|\mu|$ is defined as*

$$|\mu|(E) = \sup \left\{ \sum_{n=0}^\infty |\mu(E_n)| \;\middle|\; E_n \in \mathcal{B}(\Omega) \textit{ pairwise disjoint}, \, E = \bigcup_{n=0}^\infty E_n \right\}$$

Definition 2.5. *The **space of Radon measures** is the set*

$$\mathcal{M}(\Omega, \mathbb{R}^d) = \left\{ \mu \in \mathcal{B}(\Omega) \to \mathbb{R}^d) \,\middle|\, \mu \text{ Radon measure} \right\}.$$

$\mathcal{M}(\Omega, \mathbb{R}^d)$ together with the norm $\|\mu\|_{\mathcal{M}} = |\mu|(\Omega)$ is a Banach space.

The next theorem will state a well-known characterization of the space $\mathcal{M}(\Omega, \mathbb{R}^d)$.

Theorem 2.6. *Let X be a locally compact and separable metric space. Then the dual of the Banach space $C_0(X, \mathbb{R}^d)$ can be identified with $\mathcal{M}(X, \mathbb{R}^d)$ under the pairing*

$$\langle \mu, u \rangle = \int_X u \, \mathrm{d}\mu.$$

Moreover the dual norm is $|\mu|(X)$.

Proof. See [2] Remark 1.57. □

Hence any Radon measure can be identified by a linear functional acting on continuous functions. We close this section with a list of convenient properties that will be used throughout this paper.

Theorem 2.7.

1. *Let $\mu : \mathcal{B}(\Omega) \to \mathbb{R}^d$ be a finite Radon measure. Then its total variation measure is always a positive finite Radon measure, i.e. $|\mu| \geq 0$.*

2. *For any function $u \in \mathrm{BV}(\Omega)$ its distributional derivative ∇u is a Radon measure in $\mathcal{M}(\Omega, \mathbb{R}^2)$.*

3. *For any function $u \in \mathrm{BV}(\Omega)$ its distributional derivative ∇u can be decomposed with respect to its total variation measure, i.e. there exists a density $\sigma \in L^1_{|\nabla u|}(\Omega, \mathbb{R}^2)$ such that*

$$\nabla u = \sigma \, |\nabla u|$$

where $\sigma(x) \in S^1$ $|\nabla u|$-almost everywhere. The pairing $(\sigma, |\nabla u|)$ is called polar decomposition of ∇u.

Proof. We will not exercise proofs of these statements here, since they can for example be found in [2]. For the first and last statement see [2, Theorem 1.6 and Corollary 1.29]. For the second statement notice that the definition of $\mathrm{BV}(\Omega)$ used in [2] differs from the one used in this thesis, as it defines $u \in L^1(\Omega)$ to be a function of bounded variation if the distributional derivative of u is representable by a finite Radon measure, see [2, Definition 3.1]. However by [2, Proposition 3.6] (Variation of BV functions) the two definitions are compatible. □

2.1.2 Distance functions on measures

In this section we will introduce two distance functions on $\mathcal{M}(\Omega)$ that will later be used in a specialized type of noise removal.

Wasserstein distance

The first metric in consideration is the Wasserstein metric, which is commonly used in optical transport problems. The idea is to compare two measures of the same mass by finding an optimal plan to transform one into the other using translations. The cost associated with the transformation is defined through the distance function of the metric space in consideration. Hence, the greater the distance, the more expensive the transformation is. Precisely:

Definition 2.8. *Let (X, d) be a complete separable metric space and $p \geq 1$. The **Wasserstein-p distance** between two probability measures $\mu^1, \mu^2 \in \mathcal{P}_p(X)$ is defined by*

$$\operatorname{dist}_{W_p}^p(\mu^1, \mu^2) = \inf \left\{ \int_{X^2} d(x_1, x_2)^p \, \mathrm{d}\mu(x_1, x_2) \;\middle|\; \mu \in \Gamma(\mu^1, \mu^2) \right\}$$

where $\Gamma(\mu^1, \mu^2) = \left\{ \gamma \in \mathcal{P}(X^2) \;\middle|\; \pi^i_{\#}\gamma = \mu^i \text{ for } i = 1, 2 \right\}$ is the set of transport plans, $\pi^i_{\#} : X^2 \to X$ denotes the canonical projections, and $\mathcal{P}_p$ is the set of all probability measures with finite p-th moment, i.e. $\mathcal{P}_p(X) = \{\mu \in \mathcal{P}(X) \mid \int_X d(x, \bar{x})^p \, \mathrm{d}\mu(x) < +\infty \text{ for some } \bar{x} \in X\}$.

We will only utilize the Wasserstein distance for $p = 1$, which can be expressed in the following dual formulation.

Theorem 2.9. *The Wasserstein-1 distance for two probability measures $\mu^1, \mu^2 \in \mathcal{P}(X)$ with bounded support can be expressed as*

$$\operatorname{dist}_{W_1}(\mu^1, \mu^2) = \sup \left\{ \int_X \varphi \, \mathrm{d}(\mu^1 - \mu^2) \;\middle|\; \varphi : X \to \mathbb{R} \;\; 1 - \text{Lipschitz} \right\}.$$

Proof. See [3] Section 7.1. □

Remark 2.10. *Notice that $\operatorname{dist}_{W^1}(\mu^1, \mu^2)$ can only be finite if $(\mu^1 - \mu^2)(X) = \int_X 1 \, \mathrm{d}(\mu^1 - \mu^2) = 0$, since constant functions are 1-Lipschitz. Moreover $\operatorname{dist}_{W^1}$ corresponds to the dual Lipschitz norm and hence is convex.*

$\mathbf{H^{-1}}$ distance

The second metric in consideration is a distance measured in H^{-1}, i.e. the space of bounded linear functionals on the Sobolev space H^1_0. We will give a characterization of the H^{-1} norm that is closely related to Poisson's equation. In some sense, we therefore compare two measures by smoothing their differences and computing the total energy.

Throughout this section $\Omega \subset \mathbb{R}^n$ denotes an open and bounded domain.

Definition 2.11. *Denote by $H^1(\Omega) = W^{1,2}(\Omega)$ is the Sobolev space and by H^1_0 the closure of $C^\infty_c(\Omega)$ in $H^1_0(\Omega)$ endowed with the norm*

$$\|f\|_{H^1_0} = \|\nabla f\|_{L^2(\Omega)} \quad \text{for } f \in H^1_0.$$

Definition 2.12. *$H^{-1}(\Omega)$ denotes the dual space to $H_0^1(\Omega)$ endowed with the norm*

$$\|f\|_{H^{-1}} = \sup_{u \in H_0^1} \frac{\langle f, u\rangle_{H^{-1},H_0^1}}{\|u\|_{H_0^1}} \quad \textit{for } f \in H^{-1},$$

where the dual pairing is defined as $\langle f, u\rangle_{H^{-1},H_0^1} = \int_\Omega f u \,\mathrm{d}x$.

Theorem 2.13. *Let $\Omega \subset \mathbb{R}^n$ be a bounded and open set and $g \in H^{-1}(\Omega)$, then*

$$\|g\|_{H^{-1}} = \left\|\nabla(-\Delta)^{-1} g\right\|_{L^2}.$$

Proof. Let $g \in H^{-1}(\Omega)$. Using Riesz's representation theorem we can deduce the existence of an isometric isomorphism $\Phi : H^{-1} \to H_0^1$ such that $\langle g, v\rangle_{H^{-1},H_0^1} = \langle \Phi(g), v\rangle_{H_0^1}$ for all $v \in H_0^1$ and

$$\|g\|_{H^{-1}} = \sup_{f \in H_0^1} \frac{\langle g, f\rangle_{H^{-1},H_0^1}}{\|f\|_{H_0^1}} = \sup_{f \in H_0^1} \frac{\langle \Phi(g), f\rangle_{H_0^1}}{\|f\|_{H_0^1}} = \|\Phi(g)\|_{H_0^1} = \|\nabla\Phi(g)\|_{L^2} \tag{i}$$

since the optimum is attained at $\Phi(g)$. Moreover,

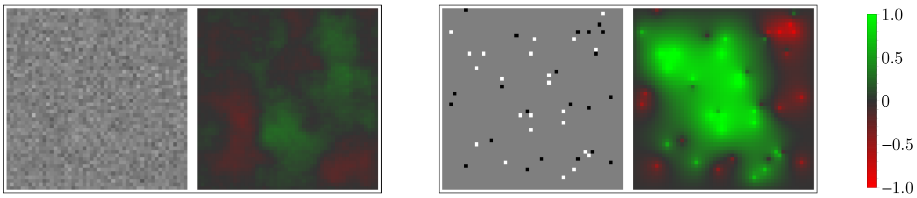
$$\langle g, f\rangle_{H^{-1},H_0^1} = \int_\Omega g f \,\mathrm{d}x = \langle \Phi(g), f\rangle_{H_0^1} = \int_\Omega \nabla\Phi(g)\,\nabla f \,\mathrm{d}x \quad \text{for all } f \in H_0^1$$

which coincides with the weak formulation of Poisson's equation with zero boundary conditions, and consequently $\Phi(g) = (-\Delta)^{-1} g$. Together with (i) this establishes the statement. □

Using this theorem we can define a distance function between measures.

Definition 2.14. *Let $\mu^1, \mu^2 \in \mathcal{M}(\Omega) \cap H^{-1}(\Omega)$. Then*

$$\operatorname{dist}^2_{H^{-1}}(\mu^1, \mu^2) = \left\|\mu^1 - \mu^2\right\|^2_{H^{-1}} = \left\|\nabla(-\Delta)^{-1}(\mu^1 - \mu^2)\right\|^2_{L^2}.$$



(a) *Gaussian noise,* $\sigma = 0.05$

(b) *Salt-and-pepper noise,* 1% *corrupted*

Figure 2.1: *Comparison of two noise profiles and the solution of Poisson's equation. Observe that the energy required to remove the salt-and-pepper noise is much higher, although only few noise particles have been introduced. Hence we expect that the H^{-1} metric will be better suited for Gaussian noise removal, since removing single particles with strong peaks is associated with very high costs.*

2.2 Convex optimization

To compute numerical results we will propose different algorithms that each require optimization problems to be stated in a standard form. The next chapter will show that, for example linear programming is well suited for many problems using the proposed curvature regularization. In its standard form it can only handle minimization or maximization problems while the regularization framework consists of a convex-concave saddle-point form. To overcome this limitation we will apply classic results from duality theory.

We begin with a few standard definitions. This section based on [12].

Definition 2.15. *A set $S \subset \mathbb{R}^n$ is said to be*

convex *if $\lambda x + (1-\lambda)y \in S$ whenever $x, y \in S$ and $0 \leq \lambda \leq 1$,*

affine *if $\lambda x + (1-\lambda)y \in S$ whenever $x, y \in S$ and $\lambda \in \mathbb{R}$.*

*The **affine hull** of S is the smallest affine set containing S, i.e.*

$$\operatorname{aff} S = \left\{ y = \sum_{i=1}^{k} \lambda_i x_i \;\middle|\; k \in \mathbb{N},\ \lambda_i \in \mathbb{R},\ x_i \in S \text{ for } 1 \leq i \leq k,\ \sum_{i=1}^{k} \lambda_i = 1 \right\}.$$

Next we will require a definition for the interior of a convex set. However, often we are dealing with low-dimensional subsets in a higher dimensional space and therefore the topological interior is not meaningful.

Definition 2.16. *Let $C \subset \mathbb{R}^n$ be a non-empty convex set. Then the **relative interior** of C is defined as* $\operatorname{relint} C = \{x \in \operatorname{aff} C \mid \exists \epsilon > 0, B_\epsilon(x) \cap \operatorname{aff} C \subset C\}$.

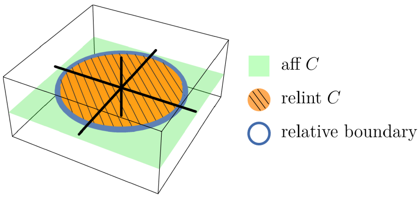


Figure 2.2: *Relative interior for* $C = \left\{x \in \mathbb{R}^3 \;\middle|\; x_1^2 + x_2^2 \leq 1, x_3 = 0\right\}$.

Definition 2.17. *For a function $f : S \subset \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$ the **epigraph** is the set*

$$\operatorname{epi} f = \{(x, \lambda) \mid x \in S, \lambda \in \mathbb{R}, \lambda \geq f(x)\}$$

and f is said to be convex, if $\operatorname{epi} f$ *is a convex subset in $\mathbb{R}^{n+1}$. Further f is said to be concave, if $-f$ is convex.*

Definition 2.18. *For a set $S \in \mathbb{R}^n$ its **indicator function** is defined as*

$$\mathbb{1}_S^\infty(x) = \begin{cases} 0 & \text{if } x \in S, \\ \infty & \text{otherwise.} \end{cases}$$

Clearly a set is convex if and only if its associated indicator function is convex.

Definition 2.19. *Let $f : S \subset \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$ be a convex function. Then the* ***effective domain*** *of f is defined as the set*

$$\operatorname{dom} f = \{x \in \mathbb{R}^n \mid \exists \lambda,\ (x, \lambda) \in \operatorname{epi} f\} = \{x \mid f(x) < +\infty\}\,.$$

The convex function f is said to be ***proper (convex)****, if its epigraph is non-empty and contains no vertical lines, i.e. if $f(x) < +\infty$ for at least one x and $f(x) > -\infty$ for every x. Likewise, a function g is said to be* ***proper concave****, if $-g$ is proper (convex).*

Another way to check if a convex function f is proper is to inspect the effective domain, i.e. f is proper if and only if $\operatorname{dom} f$ is non-empty and f restricted to $\operatorname{dom} f$ is finite.

Definition 2.20. *A function $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is said to be* ***closed*** *if $\operatorname{epi} f$ is closed.*

For convex functions we can find a more advanced characterization for closedness that directly relates to lower semi-continuity.

Theorem 2.21. *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$. Then the following conditions are equivalent:*

(a) f is lower semi-continuous, i.e. $\liminf_{x \to x_0} f(x) \geq f(x_0)$ for all $x_0 \in \mathbb{R}^n$.

(b) $\operatorname{epi} f$ is a closed set in $\mathbb{R}^{n+1}$.

Proof. See [12, Theorem 7.1]. □

Definition 2.22. *The* ***convex relaxation*** *or* ***lower semi-continuous envelope*** *of a proper convex function f is defined as*

$$\bar{f}(x) = (\operatorname{cl} f)(x) = \liminf_{y \to x} f(y).$$

In other words, the lower semi-continuous envelope is the greatest lower semi-continuous function that is majorized by f. With that definition we can give an alternative characterization for closedness of a convex function. A convex function f is closed if $\operatorname{cl} f = f$. If f is proper then closedness is the same as lower semi-continuity.

Now with this short summary of all requirements we can formulate the main result of this section.

Definition 2.23. *The* ***conjugate*** *of a function $f : \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$ is defined as*

$$f^*(x^*) = \sup_{x \in \mathbb{R}^n} x \cdot x^* - f(x).$$

If f is concave then its ***concave conjugate*** *is defined as*

$$f^\star(x^*) = -f^*(-x^*) = \inf_{x \in \mathbb{R}^n} x \cdot x^* - f(x).$$

Theorem 2.24 (Fenchel's Duality Theorem)**.** *Let f be a proper convex function on $\mathbb{R}^n$, and let g be a proper concave function on $\mathbb{R}^n$. One has*

$$\inf_x f(x) - g(x) = \sup_{x^*} g^\star(x^*) - f^*(x^*)$$

if either of the following conditions is satisfied:

(a) $\operatorname{relint} \operatorname{dom} f \cap \operatorname{relint} \operatorname{dom} g \neq \emptyset$*,*

(b) f and g are closed, and $\operatorname{relint} \operatorname{dom} g^\star \cap \operatorname{relint} \operatorname{dom} f^* \neq \emptyset$*.*

Under (a) the supremum is attained at some x^*, while under (b) the infimum is attained at some x. If (a) and (b) both hold, the infimum and supremum are necessarily finite.

Proof. See [12, Theorem 31.1]. □

Theorem 2.25. *Let f be a closed proper convex function on $\mathbb{R}^n$, let g be a closed proper concave function on $\mathbb{R}^m$, and let $A : \mathbb{R}^n \to \mathbb{R}^m$ be a linear operator. One has*

$$\inf_x f(x) - g(Ax) = \sup_{x^*} g^\star(x^*) - f^*(A^T x^*)$$

if either of the following conditions is satisfied:

(a) There exists an $x \in$ $\operatorname{relint} \operatorname{dom} f$ *such that $Ax \in$* $\operatorname{relint} \operatorname{dom} g$*,*

(b) There exists an $x^* \in$ $\operatorname{relint} \operatorname{dom} g^\star$ *such that $A^T x^* \in$* $\operatorname{relint} \operatorname{dom} f^*$*.*

Under (a) the supremum is attained at some x^*, while under (b) the infimum is attained at some x.

Proof. See [12, Corollary 31.2.1]. □

The following proposition is of course an application of Theorem 2.25 and will be helpful in recasting optimization problems in a saddle-point form to a minimization problem, especially when the linear objective function is subject to a Lipschitz constraint.

Proposition 2.26. *For $\beta \in \mathbb{R}^n$ and $C \in \mathbb{R}^{m \times n}$ the following equality holds:*

$$\sup_{\substack{\varphi \in \mathbb{R}^n \\ \|C\varphi\|_\infty \le 1}} \varphi \cdot \beta = \min_{\substack{\varphi^* \in \mathbb{R}^m \\ C^T\varphi^* + \beta = 0}} \|\varphi^*\|_1 .$$

Proof. The above problem can be reformulated as

$$\sup_{\substack{\varphi \in \mathbb{R}^n \\ \|C\varphi\|_\infty \le 1}} \varphi \cdot \beta = - \inf_{\substack{\varphi \in \mathbb{R}^n \\ \|C\varphi\|_\infty \le 1}} -\varphi \cdot \beta = - \inf_{\varphi \in \mathbb{R}^n} -\varphi \cdot \beta - \big(- \mathbb{1}^{\infty}_{\{\|\widehat{\varphi}\|_\infty \le 1\}}(C\varphi)\big)$$

which motivates the definition of $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^m \to \mathbb{R} \cup \{-\infty\}$ as

$$f(\varphi) = -\varphi \cdot \beta, \qquad g(\varphi) = -\mathbb{1}^{\infty}_{\{\|\widehat{\varphi}\|_\infty \le 1\}}(\varphi).$$

As f is a linear function it is convex, proper and continuous and therefore, using Theorem 2.21, also closed. Obviously g is proper concave and closed since its epigraph can be computed as $\operatorname{epi} -g = \{(x, \lambda) \mid \|x\|_\infty \le 1, \lambda \ge 0\} = [-1,1]^m \times [0, \infty)$ and hence is closed in $\mathbb{R}^{m+1}$. Moreover

$$\begin{aligned} \operatorname{dom} f = \{\varphi \mid -\varphi \cdot \beta < +\infty\} = \mathbb{R}^n \quad &\Rightarrow \quad \operatorname{relint} \operatorname{dom} f = \mathbb{R}^n, \\ \operatorname{dom} g = \{\widehat{\varphi} \mid -g(\widehat{\varphi}) < +\infty\} = \{\widehat{\varphi} \mid \|\widehat{\varphi}\|_\infty \le 1\} \quad &\Rightarrow \quad \operatorname{relint} \operatorname{dom} g = \{\widehat{\varphi} \mid \|\widehat{\varphi}\|_\infty < 1\}\,. \end{aligned}$$

Trivially $0 \in \operatorname{relint} \operatorname{dom} f$ and $C^T 0 = 0 \in \operatorname{relint} \operatorname{dom} g$. Hence requirement (a) of Theorem 2.25 is met and we have

$$\sup_{\substack{\varphi \in \mathbb{R}^n \\ \|C\varphi\|_\infty \le 1}} \varphi \cdot \beta = -\inf_{\varphi \in \mathbb{R}^n} f(\varphi) - g(C\varphi) = -\max_{\widehat{\varphi} \in \mathbb{R}^m} g^\star(\widehat{\varphi}) - f^*(C^T \widehat{\varphi}) = \min_{\widehat{\varphi} \in \mathbb{R}^m} f^*(C^T \widehat{\varphi}) - g^\star(\widehat{\varphi})$$

and the conjugates are computed as

$$f^*(\varphi^*) = \sup_{\varphi \in \mathbb{R}^n} \varphi \cdot \varphi^* + \beta \cdot \varphi = \begin{cases} 0 & \text{if } \varphi + \beta = 0, \\ \infty & \text{otherwise.} \end{cases}, \qquad g^\star(\varphi^*) = \inf_{\substack{\varphi \in \mathbb{R}^m \\ \|\varphi\|_\infty \le 1}} \varphi \cdot \varphi^* = -\|\varphi^*\|_1$$

which yields the proposed formula. □

3 Image Regularization Framework

The goal of this thesis is to derive a new discretization for the curvature regularization framework proposed in [7]. But before going into details we will summarize the idea and main results from the original article. Technical results will only be stated if necessary for the new results from this thesis.

3.1 Functional lifting

Utilizing the theory from the first chapter we will next introduce the concept of functional lifting, which will be the foundation for finding a convex relaxation of a class of functionals that, when applied to images, are able to penalize curvature of the level lines. Consider the following motivational example.

Let $x : I \to \Omega \subset \mathbb{R}^2$ be the parameterization of a smooth planar curve with inner normal $\nu : I \to S^1$. Then the lifting of this curve is defined as $s \mapsto (x(s), \nu(s))$. Through the lifting the intrinsic local orientation of the curve is now accessible in the additional dimension. To apply this to level lines of images we need to generalize this idea. For one we need a mechanism to extract local orientations, and we need to keep in mind that level lines in general are non-smooth curves.

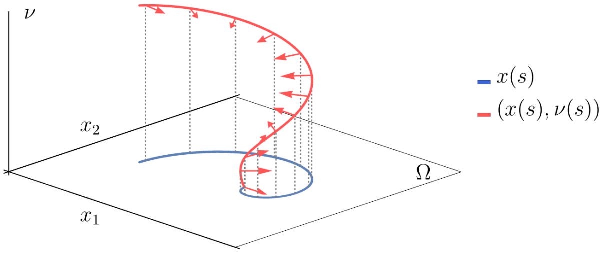


Figure 3.1: *Lifting of a smooth curve to the space $\Omega \times S^1$. Depending on the orientation of the normal, indicated by the red arrows, the curve is lifted to different heights in three-dimensional space.*

This will be addressed using the theory from the first chapter. For any image $u \in \mathrm{BV}(\Omega)$ we know that the distributional derivative ∇u is a Radon measure that can readily be decomposed using the polar decomposition from Theorem 2.7, i.e. $\nabla u = \sigma \, |\nabla u|$. As pointed out before, the density σ conveniently satisfies that $\sigma(x) \in S^1$ $|\nabla u|$-almost everywhere and is oriented normal to the level lines. Since we are considering $\Omega \subset \mathbb{R}^2$ we can easily rotate the density to be tangentially to the level lines.

Definition 3.1 (Rotation Operator $\perp$). *The operator that rotates $\nu \in S^1$ counterclockwise by $\frac{\pi}{2}$ is defined as*

$$\nu^\perp := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \nu.$$

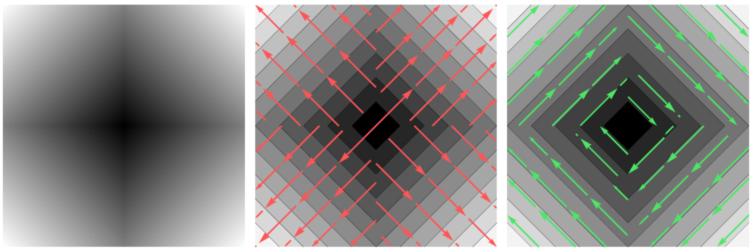


Figure 3.2: *Example for $u : \Omega = (-1,1)^2 \to \mathbb{R}, u(x) = \|x\|_1$. The first figure displays u, the second illustrates the level lines of u and the orientations of density σ, and the third figure shows the orientations of $-\sigma^\perp$. Notice that by choosing the negative rotated direction the function is increasing on the left-hand side.*

Equipped with the necessary theory we are now ready to define the functional lifting of an image gradient.

Definition 3.2 (Functional lifting of ∇u). *Let $u \in \mathrm{BV}(\Omega)$. The functional lifting of ∇u is the measure $\mu = \mu(\nabla u) \in \mathcal{M}(\Omega \times S^1)$ with*

$$\int_{\Omega \times S^1} \varphi \, \mathrm{d}\mu = \int_\Omega \varphi(x, -\sigma(x)^\perp) \, \mathrm{d}\, |\nabla u|$$

for each $\varphi \in C_0(\Omega \times S^1)$.

This definition and its properties are best explained on a simple example. Consider $u \in \mathrm{BV}(\Omega)$ as a binary image of a polygon P as in Figure 3.3. The distributional derivative ∇u is of course constantly zero inside and outside the polygon and has support only on the boundary $\Gamma = \cup_i \Gamma_i$ where u jumps from 0 to 1.

As we decompose ∇u on Γ to $\nabla u = \sigma \, |\nabla u|$ we notice that the density $\sigma(x)$ is pointing to the interior of the polygon. On each piece Γ_i of the boundary $\sigma(x)$ is constantly pointing in the same direction. Consequently when we apply the functional lifting, each piece of the boundary is lifted to different heights, where the height is depending only on the orientation of $\sigma(x)$.

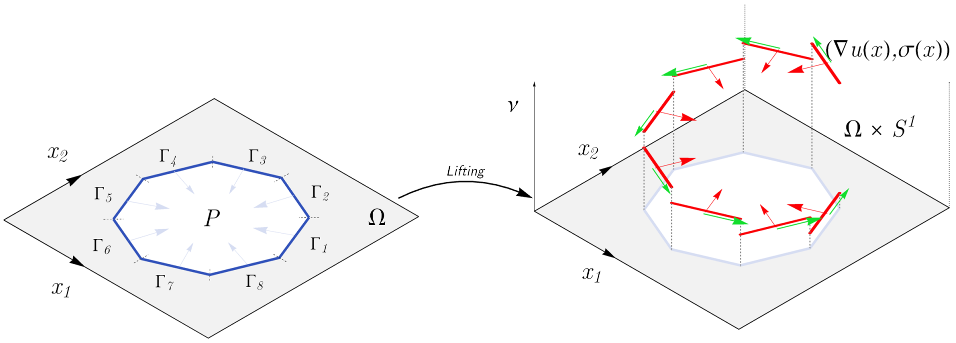


Figure 3.3: *Lifting applied on a simple polygonal binary image u, where u is constant inside and outside of the polygon P, i.e. $u|_P = 1$ and $u|_{\Omega\setminus P} = 0$. Notice again that $-\sigma(x)^\perp$, indicated by the green arrows, is oriented so that the function is increasing on the left-hand side.*

The measure μ associated with ∇u now acts on these lifted level lines. Applied to a function $\varphi \in C_0(\Omega \times S^1)$ it integrates only along the corresponding tangential direction. Further notice the following properties of μ:

1. $x \mapsto (x, -\sigma(x)^\perp)$ is measurable (with respect to $|\nabla u|$) between $\Omega \to \Omega \times S^1$.
2. μ is a positive Radon measure.
3. The measure $|\nabla u|$ can be recovered from μ by

$$\int_\Omega \varphi \,\mathrm{d}\,|\nabla u| = \int_{\Omega\times S^1} \varphi(x)\,\mathrm{d}\mu(x,\nu) \qquad \text{for all } \varphi \in C_0(\Omega).$$

4. In the same way the measure ∇u can be recovered from μ by

$$\int_\Omega \varphi \cdot \,\mathrm{d}\,\nabla u = \int_\Omega \varphi(x) \cdot \underbrace{\sigma(x)}_{-\sigma^{\perp\perp}} \,\mathrm{d}\,|\nabla u| = \int_{\Omega\times S^1} \varphi(x)\sigma(x)^\perp \,\mathrm{d}\mu \quad \text{for all } \varphi \in C_0(\Omega, \mathbb{R}^2).$$

In the next section we will continue with the example of a binary polygonal image. The goal still is to derive functionals that are able to penalize vertices of a polygon. However, instead of trying to solve this task for the flat two-dimensional image we can now directly utilize the orientations of the gradient in $\Omega \times S^1$.

3.2 Functionals acting on polygons

This section will introduce functionals that are able to penalize vertices of characteristic functions of polygons by acting on the lifted gradient. As a first motivational example consider a polygon consisting only of a single vertex centered at zero. Precisely:

Let $\Omega = B_1(0)$ be the (open) unit disk centered at zero. Further let $\mu \in \mathcal{M}(\Omega \times S^1)$ represent the integration on two unit line segments with orientations $\nu_1, \nu_2 \in S^1$ meeting at zero. Then for each $\varphi \in C_0(\Omega \times S^1)$

$$\int_{\Omega\times S^1} \varphi \,\mathrm{d}\mu = \int_{-1}^{0} \varphi(t\nu_1, \nu_1)\,\mathrm{d}t + \int_0^1 \varphi(t\nu_2, \nu_2)\,\mathrm{d}t.$$

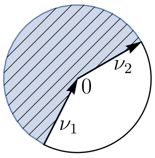


Figure 3.4: *Single vertex polygon in $B_1(0)$ with incoming and outgoing orientations ν_1 and ν_2.*

Now we consider the distributional directional derivative of μ with respect to $(-\nu, 0)$, i.e. we test with $\varphi = \nabla_x \psi(x,\nu) \cdot \nu$ for $\psi \in C_c^\infty(\Omega \times S^1)$:

$$\begin{aligned}\int_{\Omega \times S^1} \nabla_x \Psi(x,\nu) \cdot \nu \, \mathrm{d}\mu &= \int_{-1}^{0} \frac{\partial}{\partial t} \Psi(t\nu_1, \nu_1) \, \mathrm{d}t + \int_0^1 \frac{\partial}{\partial t} \Psi(t\nu_2, \nu_2) \, \mathrm{d}t \\ &= \Psi(0,\nu_1) - \Psi(0,\nu_2) \\ &= \langle \delta_{\nu_1} - \delta_{\nu_2}, \Psi(0,\cdot) \rangle\end{aligned}$$

where δ_ν denotes the delta distribution at ν. This result leads to the following observations:

1. In order to penalize vertices of polygons, we need functionals acting on $\mathcal{M}(S^1)$.
2. It is desirable to control the penalty for each pair of orientations ν_1 and ν_2 individually.

To measure differences between orientation we can of course always select a metric on S^1. The idea is then to generalize this metric to act on $\mathcal{M}(S^1)$ which is possible for the following class of metrics.

Assumption 3.3. *Let $\rho : S^1 \times S^1 \to [0,\infty)$ such that*

1. *ρ defines a metric on S^1*
2. *ρ is lower semi-continuous, i.e. $\rho = \sup_{i \in I} \rho_i$ where I is a non-empty index set and $\rho_i \in C(S^1 \times S^1)$ is a continuous metric for each $i \in I$.*

Definition 3.4. *For a metric $\rho : S^1 \times S^1 \to [0,\infty)$ satisfying Assumption 3.3 the functional $\|\cdot\|_\rho : \mathcal{M}(S^1) \to [0,\infty]$ is defined as*

$$\|\mu\|_\rho = \sup_{\varphi \in C_\rho} \langle \mu, \varphi \rangle$$

with $C_\rho = \{\varphi \in C(S^1) \mid \varphi(\eta_1) - \varphi(\eta_2) \leq \rho(\eta_1, \eta_2) \text{ for all } \eta_1, \eta_2 \in S^1\}$.

Definition 3.5. *We call a functional a weak* sequentially lower semi-continuous norm, if it is*

1. *non-negative,*
2. *positively homogeneous and positive definite,*
3. *weak* sequentially lower semi-continuous,*
4. *satisfies the triangle inequality,*
5. *but may also attain the value ∞.*

The following theorem will show that $\|\cdot\|_\rho$ indeed satisfies Definition 3.5.

Theorem 3.6. *The functional $\|\cdot\|_\rho : \mathcal{M}(S^1) \to [0,\infty]$ has the following properties:*

1. *$\|\cdot\|_\rho$ is a weak* sequentially lower semi-continuous norm,*

2. *for each $\nu_1, \nu_2 \in S^1$, it holds that $\|\delta_{\nu_1} - \delta_{\nu_2}\|_\rho = \rho(\nu_1, \nu_2)$.*

Proof. From the symmetry of ρ we conclude that $\varphi \in C_\rho \Leftrightarrow -\varphi \in C_\rho$ and therefore $C_\rho = -C_\rho$. Hence we obtain the non-negativity

$$\|\mu\|_\rho = \sup_{\varphi \in C_\rho} \langle \mu, \varphi \rangle = \sup_{\varphi \in C_\rho} |\langle \mu, \varphi \rangle| \geq 0.$$

Using the same argument and with $\lambda \in \mathbb{R}$ we can establish the positive homogeneity

$$\|\lambda\mu\|_\rho = \sup_{\varphi \in C_\rho} \langle \lambda\mu, \varphi \rangle = \sup_{\varphi \in C_\rho} \langle |\lambda| \, \mu, \operatorname{sgn}(\lambda)\varphi \rangle = |\lambda| \sup_{\operatorname{sgn}(\lambda) C_\rho} \langle \mu, \varphi \rangle = |\lambda| \, \|\mu\|_\rho \, .$$

Since $\|\cdot\|_\rho$ is the pointwise supremum of sequentially weak* continuous functionals, the weak* lower semi-continuity follows directly. Now for all $\mu_1, \mu_2 \in \mathcal{M}(S^1)$ and $\varphi \in C_\rho$ we have

$$\langle \mu_1 + \mu_2, \varphi \rangle \leq \|\mu_1\|_\rho + \|\mu_2\|_\rho \Rightarrow \|\mu_1 + \mu_2\|_\rho = \sup_{\varphi \in C_\rho} \langle \mu_1 + \mu_2, \varphi \rangle \leq \|\mu_1\|_\rho + \|\mu_2\|_\rho$$

which yields the triangle inequality. As the proof of the positive definiteness and the second statement requires more work we refer to [7, Proposition 3.5]. □

Remark 3.7. *Since C_ρ contains constant functions, $\|\mu\|_\rho$ is only finite if $\int_{S^1} 1 \, \mathrm{d}\mu = 0$. Otherwise:*

$$\sup_{\varphi \in C_\rho} \langle \mu, \varphi \rangle \geq \sup_{c \in \mathbb{R}} \int_{S^1} c \, \mathrm{d}\mu = \infty.$$

As stated in the introduction we will consider two metrics. The first one will turn out to count the number of vertices with distinct incoming and outgoing orientations and the other to measure the total sum of absolute exterior angles between incoming and outgoing edges.

Example 3.8 (Discrete Metric)**.** *Let $\rho_0 : S^1 \times S^1 \to \{0, 1\}$ be the discrete metric, i.e.*

$$\rho_0(\nu_1, \nu_2) = \begin{cases} 0 & \textit{if } \nu_1 = \nu_2 \\ 1 & \textit{if } \nu_1 \neq \nu_2 \end{cases} = \sup_{\lambda > 0} \min(1, \lambda^{-1} |\nu_1 - \nu_2|).$$

This metric is admissible in the sense of Assumption 3.3. The set C_{ρ_0} contains all functions $\varphi \in C(S^1)$ that satisfy $\max_{\nu_1 \in S^1} \varphi(\nu_1) - \min_{\nu_2 \in S^1} \varphi(\nu_2) \leq 1$. This can be restated as

$$C_{\rho_0} = \mathbf{1}\mathbb{R} + \{\|\varphi\|_\infty \leq \tfrac{1}{2}\} = \mathbf{1}\mathbb{R} + \overline{\{\varphi \in C^\infty(S^1) \mid \|\varphi\|_\infty \leq \tfrac{1}{2}\}},$$

where the closure is being taken in $C(S^1)$, and $\mathbf{1}$ denotes the constant 1-function Hence for $\mu \in \mathcal{M}(S^1)$ and in light of Remark 3.7 we can compute $\|\mu\|_{\rho_0}$ as

$$\|\mu\|_{\rho_0} = \begin{cases} \sup_{\|\varphi\|_\infty \leq \frac{1}{2}} \langle \mu, \varphi \rangle = \frac{1}{2} \|\mu\|_{\mathcal{M}} & \textit{if } \int_{S^1} 1 \, \mathrm{d}\mu = 0, \\ \infty & \textit{else.} \end{cases}$$

Example 3.9 (Geodesic metric)**.** *Let $\rho_1 : S^1 \times S^1 \to [0, \pi]$ be the metric measuring geodesic distances, i.e.*

$$\rho_1(\nu_1, \nu_2) = \min \left\{ |t_1 - t_2| \mid \nu_i = \big(\cos(t_i), \sin(t_i)\big),\ i = 1, 2 \right\}.$$

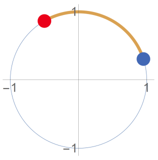


Figure 3.5: *Geodesic distance on S^1.*

Then the set C_{ρ_1} consists of all Lipschitz continuous functions in $C(S^1)$ with Lipschitz constant not exceeding 1, i.e.

$$\begin{aligned} C_{\rho_1} &= \left\{ \varphi \in C(S^1) \,\middle|\, \varphi'(\nu) \text{ exists for almost every } \nu \in S^1 \text{ and } \|\varphi'\|_\infty \le 1 \right\} \\ &= \overline{\{\varphi \in C^\infty(S^1) \mid \|\varphi'\|_\infty \le 1\}} \end{aligned}$$

where the closure again is being taken in $C(S^1)$. The functional $\|\cdot\|_{\rho_1}$ then corresponds to a dual Lipschitz norm.

3.3 Generalization to admissible polygons

We will continue by generalizing the single vertex polygon case to a general class of admissible polygons.

Definition 3.10 (Admissible polygons)**.** *Let $\Omega \subset \mathbb{R}^2$ be a bounded Lipschitz domain. A relatively closed polygon $P \subset \Omega$ is said to be an admissible polygon if*

1. *the line segments of P in Ω are (up to boundary points) given by*

   $$[x_i, y_i] = \{\lambda x_i + (1 - \lambda) y_i \mid \lambda \in [0, 1]\}$$

   for $I \ge 3$ and $x_1, ..., x_I, y_1, ..., y_I \in \overline{\Omega}$,
2. *the vertices are distinct, i.e. the collections $(x_i)_{1 \le i \le I}, (y_i)_{1 \le i \le I}$ are each pairwise disjoint,*
3. *the line segments are connected, i.e. for $1 \le i \le I$ it holds that $x_{i+1} = y_i$ if $y_i \in \Omega$ and that y_i and x_{i+1} lie on the same connected component of $\partial\Omega$ if $y_i \in \partial\Omega$, where $x_{I+1} = x_1$,*
4. *the line segments $[x_i, y_i]$ are pairwise disjoint for $1 \le i \le I$, i.e., the polygon does not intersect itself,*
5. *P lies on the left hand side with respect to the oriented segments $[x_i, y_i]$, i.e. for each $x = \lambda x_i + (1 - \lambda) y_i$ with $1 \le i \le I$ and $\lambda \in (0, 1)$ there exists a neighborhood of U of x such that*

   $$P \cap U = \left\{ z \in U \,\middle|\, (z - x) \cdot (y_i - x_i)^\perp \ge 0 \right\}.$$

Informally speaking, a polygon P is admissible if it consists only of unique vertices, is non self-intersecting (often called a ”simple polygon”) with the extension that it can also be defined on the boundary of the potentially open set Ω. The exclusion of self-intersecting polygons further ensures that for admissible polygons the interior and exterior can be clearly distinguished by examining if the vertices are defined clockwise or counterclockwise.

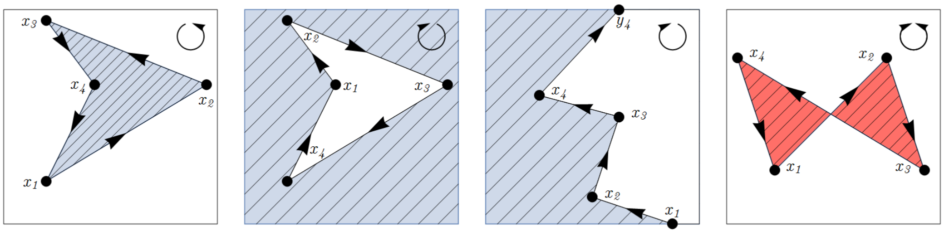


Figure 3.6: *The first three polygons are admissible and the interior is determined by the order of the vertices. The last polygon cannot be oriented and therefore it is not admissible. Note that the end points y_i are only marked if they do not correspond to x_{i+1}.*

Now let P be an admissible polygon, and $u = \mathbb{1}_P^\infty$ the characteristic function of P, which is of course contained in $\mathrm{BV}(\Omega)$. In light of the introductory example of a single vertex polygon, we will again compute the distributional directional derivative of the functional lifting of ∇u. We start by applying ∇u to a test function $\varphi \in C_0(\Omega, \mathbb{R}^2)$, which yields

$$\int_\Omega \varphi \, \mathrm{d}\, \nabla u = \int_\Omega \varphi \cdot \sigma \, \mathrm{d}\, |\nabla u| = \sum_{i=1}^{I} \int_0^1 |y_i - x_i| \, \varphi(x_i(t)) \cdot \nu_i^\perp \, \mathrm{d}t$$

with $x_i(t) = (1-t)x_i + t y_i, \nu_i = \frac{y_i - x_i}{|y_i - x_i|}$ for $1 \leq i \leq I$. Hence the functional lifting of ∇u as in Definition 3.2 can be computed as

$$\int_{\Omega \times S^1} \varphi \, \mathrm{d}\mu = \sum_{i=1}^{I} \int_0^1 |y_i - x_i| \, \varphi(x_i(t), \nu_i) \, \mathrm{d}t$$

for all test functions $\varphi \in C_0(\Omega \times S^1)$. This is in compliance with the notion that the functional lifting integrates tangentially along the level sets of u. Now as with the single vertex polygon we consider $\varphi(x, \nu) = \nabla_x \psi(x, \nu) \cdot \nu$ for a continuous function $\psi \in C_0(\Omega \times S^1)$ for which $\nabla_x \psi \in C_0(\Omega \times S^1, \mathbb{R}^2)$. Tested against the functional lifting μ we conclude, utilizing $\nu_{I+1} = \nu_i$,

$$\begin{aligned} \int_{\Omega \times S^1} \varphi \, \mathrm{d}\mu &= \sum_{i=1}^{I} \int_0^1 |y_i - x_i| \, \nabla_x \Psi(x(t), \nu_i) \cdot \nu_i \, \mathrm{d}t \\ &= \sum_{i=1}^{I} \int_0^1 \frac{\partial}{\partial t} \Psi(x_i(t), \nu_i) \, \mathrm{d}t \\ &= \sum_{i=1}^{I} \psi(y_i, \nu_i) - \psi(x_i, \nu_i) = \sum_{\substack{1 \leq i \leq I \\ y_i \in \Omega}} \psi(y_i, \nu_i) - \psi(y_i, \nu_{i+1}). \end{aligned}$$

If we again choose a metric ρ according to Assumption 3.3 and select ψ such that $\psi(x, \cdot) \in C_\rho$ for each $x \in \Omega$ we can establish the inequality

$$\int_{\Omega \times S^1} \varphi \, \mathrm{d}\mu \leq \sum_{\substack{1 \leq i \leq I \\ y_i \in \Omega}} \rho(\nu_i, \nu_{i+1}).$$

In Theorem 3.6 we were able to show that, for the simplified case of a single vertex polygon, the right hand side can be attained, which motivates the following definition.

Definition 3.11. *For $\rho : S^1 \times S^1 \to [0, \infty)$ according to Assumption 3.3 and $\mu \in \mathcal{M}(\Omega \times S^1)$, let*

$$T_\rho(\mu) = \sup_{\Psi \in M_\rho(\Omega)} \int_{\Omega \times S^1} \nabla_x \Psi(x, \nu) \cdot \nu \, \mathrm{d}\mu(x, \nu)$$

where $M_\rho(\Omega) = \left\{ \Psi \in C_0(\Omega \times S^1) \;\middle|\; \nabla_x \Psi \in C_0(\Omega \times S^1, \mathbb{R}^2),\ \Psi(x, \cdot) \in C_\rho \text{ for all } x \in \Omega \right\}$.

Proposition 3.12. *The functional T_ρ is sequentially weak* lower semi-continuous, positively homogeneous, and satisfies the triangle inequality.*

Proof. This statement can be derived in analogy to the first part of Theorem 3.6 by replacing C_ρ with $\{(x, \nu) \mapsto \nabla_x \psi(x, \nu) \cdot \nu \mid \psi \in M_\rho(\Omega)\}$. □

Proposition 3.13. *Let ρ be a metric satisfying Assumption 3.3. For P being an admissible polygon, $u = \mathbb{1}_P^\infty$ and μ the lifting of ∇u it holds that*

$$T_\rho(\mu) = \sum_{\substack{1 \le i \le I \\ y_i \in \Omega}} \rho\left(\frac{y_i - x_i}{|y_i - x_i|}, \frac{y_{i+1} - x_{i+1}}{|y_{i+1} - x_{i+1}|}\right).$$

Proof. See [7, Proposition 3.13]. □

The last statement verifies that for admissible polygons the functional T_ρ indeed is the sum over all vertices measuring differences of the incoming and outgoing orientations using the prescribed metric ρ.

Example 3.14. *For ρ_0 the discrete metric from Example 3.8 the functional T_{ρ_0} corresponds to to the number of vertices with a change in orientation, i.e.*

$$T_{\rho_0} = \# \left\{ y_i \in \Omega \;\middle|\; \frac{y_i - x_i}{|y_i - x_i|} \neq \frac{y_{i+1} - x_{i+1}}{|y_{i+1} - x_{i+1}|} \right\}.$$

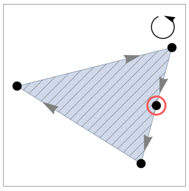


Figure 3.7: *At the marked vertex the incoming and outgoing orientations are equal. Therefore $T_{\rho_0} = 3$.*

Example 3.15. *For ρ_1 the geodesic metric from Example 3.9 we have for $y_i \in \Omega$*

$$\rho_1\left(\frac{y_i - x_i}{|y_i - x_i|}, \frac{y_{i+1} - x_{i+1}}{|y_{i+1} - x_{i+1}|}\right) = \gamma(y_i)$$

where γ is the unsigned external angle, i.e. the absolute value of the external angle between the incoming and outgoing orientation at vertex y_i. Consequently T_{ρ_1} corresponds to the sum of all external angles, i.e.

$$T_{\rho_1}(\mu) = \sum_{\substack{1\le i\le I\\ y_i\in\Omega}} \gamma(y_i).$$

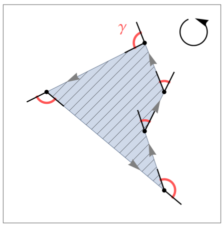


Figure 3.8: *The red arcs at each vertex indicate the unsigned external angle* $\gamma_(y_i)$.

If P *is a convex polygon with vertices completely inside* Ω*, if follows that* $T_{\rho_1} = 2\pi$.

As a final statement in this section, we will generalize the idea of polygons with straight boundaries to sets, where the vertices are connected by smooth curves. The last example already provides the intuition that T_{ρ_1} measures the total curvature of P. If we replace each corner of the polygon by an arc A_i with radius $r_i > 0$ that smoothly connects the line segments meeting at y_i, then the length of that arc would be $\gamma(y_i)r_i$ and the absolute value of the curvature $|\kappa|$ of A_i would be $\frac{1}{r_i}$. Therefore $\int_{A_i} |\kappa| \,\mathrm{d}\mathcal{H}^1 = \gamma(y_i)$.

The functional T_{ρ_0}, on the other hand, is not meaningful for sets with smooth curved boundaries since it counts the number of genuine vertices. But a set with smooth curves can be viewed as a polygon with infinitely many genuine vertices and therefore T_{ρ_0} yields ∞.

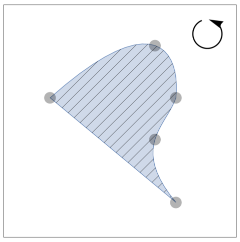


Figure 3.9: *Vertices connected by a piecewise smooth* C^2 *boundary.*

Proposition 3.16. *Let* ρ_0 *be the discrete and* ρ_1 *be the geodesic metric on* S^1*. Let* $P \subset \Omega$ *with piecewise* C^2 *boundary* $\partial P \subset \Omega$*, i.e.* ∂P *contains a set* $x_1, ..., x_I$ *of vertices which are connected by* C^2 *arcs. For simplicity, assume* ∂P *to be homeomorphic to* S^1*. Then for* μ *being the lifted gradient of the characteristic function* $\mathbb{1}_P^\infty$ *it holds that*

$$T_{\rho_0}(\mu) = \begin{cases} \#\{x_i \mid \partial P \ \textit{ is not } C^1 \textit{ at } x_i\} & \textit{if } \kappa = 0 \textit{ on } \partial P \setminus \{x_1, ..., x_I\}, \\ \infty & \textit{else}, \end{cases}$$

$$T_{\rho_1}(\mu) = \int_{\partial P \setminus \{x_1,...,x_I\}} |\kappa| \,\mathrm{d}\mathcal{H}^1 + \sum_{1\le i\le I} \gamma(x_i),$$

where κ *is the curvature of the curve* ∂P *and* $\gamma(x_i)$ *its unsigned external angle at* x_i.

Proof. See [7, Proposition 3.16]. □

3.3.1 Generalization to imaging problems

In the previous section we considered functionals T_ρ that can be applied to characteristic functions of polygons or sets with piecewise smooth boundaries. Before these functionals can be applied to image problems we need a generalization for arbitrary images. First we note that for binary $u \in \mathrm{BV}(\Omega)$ the functional lifting μ of ∇u satisfies $\mu \in \mathcal{M}(\Omega \times S^1)$, and therefore $T_\rho(\mu)$ still makes sense. We can use this observation to derive a functional that acts on the sublevel sets of u. Precisely, denoting by μ_t the functional lifting of $\nabla \mathbb{1}^\infty_{\{u<t\}}$ and $\alpha, \beta > 0$

$$R_\rho^{\alpha,\beta}(u) = \int_{\mathbb{R}} \alpha \left\|\mu_t\right\|_{\mathcal{M}} + \beta T_\rho(\mu_t) \,\mathrm{d}t. \tag{3.1}$$

This functional contains two non-linear operations that complicate numerical computation and global optimization. They can be identified as

1. the extraction of sublevel sets, i.e. $(u,t) \mapsto \mathbb{1}^{\infty}_{\{u<t\}}$,
2. the functional lifting operation, i.e. $u \mapsto \mu(\nabla u)$.

3.4 Convex Relaxation

To deal with the first problem we consider the following relaxation.

Proposition 3.17. *For $u \in \mathrm{BV}(\Omega)$, a metric ρ according to Assumption 3.3, $\alpha, \beta > 0$ and $R^{\alpha,\beta}_{\rho}$ according to* (3.1) *it holds that*

$$\alpha \left\| \mu \right\|_{\mathcal{M}} + \beta T_{\rho}(\mu) \leq R^{\alpha,\beta}_{\rho}.$$

Proof. See [7, Section 4.1]. □

That implies: Considering the functional lifting of ∇u instead of the sublevel sets gives a relaxation of $R^{\alpha,\beta}_{p}$.

As with the second problem, we consider a convex relaxation, i.e. we are looking for a convex superset of

$$G_{\nabla} = \left\{ (u,\mu) \in \mathrm{BV}(\Omega) \times \mu(\Omega \times S^1) \;\middle|\; \mu \text{ is the functional lifting of } \nabla u \right\}.$$

Although the closed convex hull of G_{∇} in an appropriate topology exists theoretically, we are looking for a superset that is computationally accessible. Therefore we notice the following minimum requirements such a superset must satisfy. Let $u \in \mathrm{BV}(\Omega)$. Then

1. the gradient lifting is a positive measure, i.e. $\mu \geq 0$,
2. for every $\varphi \in C^{\infty}_{c}(\Omega, \mathbb{R}^2)$ the following **compatibility condition** is satisfied:

$$\int_{\Omega} \varphi \cdot \nabla u \,\mathrm{d}x = \int_{\Omega} \varphi \cdot \sigma \,\mathrm{d}\left|\nabla u\right| = \int_{\Omega \times S^1} \varphi \cdot \nu^{\perp} \,\mathrm{d}\mu(x,\nu) \tag{3.2}$$

Consequently we choose

$$\begin{aligned} M_{\nabla} = \Big\{ (u,\mu) \in L^1(\Omega) \times \mathcal{M}(\Omega \times S^1) \;\Big|\; & \mu \geq 0, \\ & \int_{\Omega} \varphi \cdot \nabla u \,\mathrm{d}x = \int_{\Omega \times S^1} \varphi \cdot \nu^{\perp} \,\mathrm{d}\mu(x,\nu) \text{ for all } \varphi \in C^{\infty}_{c}(\Omega, \mathbb{R}^2) \Big\} \supset G_{\nabla} \end{aligned} \tag{3.3}$$

as a candidate for the convex relaxation, acknowledging that there may exist tighter relaxations.

Proposition 3.18. *The set M_{∇} according to* (3.3) *is non-empty, convex, and sequentially closed with respect to weak convergence in L^1 and weak* convergence in $\mathcal{M}(\Omega \times S^1)$. Moreover, for each $(u,\mu) \in M_{\nabla}$ it follows that $u \in \mathrm{BV}(\Omega)$.*

Proof. See [7, Proposition 4.1]. □

Combining these relaxations yields the following central definition:

Definition 3.19. *Let $\alpha, \beta > 0$. For $u \in L^1(\Omega)$ define*

$$\bar{R}_\rho^{\alpha,\beta}(u) = \inf_{\substack{\mu \in \mathcal{M}(\Omega \times S^1) \\ (u,\mu) \in M_\nabla}} \alpha \|\mu\|_{\mathcal{M}} + \beta T_\rho(\mu)$$

where we set the infimum of the empty set to ∞.

Remark 3.20. *The weighting parameter α controls the influence of the length-based term, while β controls the influence of the curvature term.*

Proposition 3.21. *The functional $\bar{R}_\rho^{\alpha,\beta} : L^1(\Omega) \to [0, \infty]$ is lower semi-continuous, convex, and positively one-homogeneous, i.e. $\bar{R}_\rho^{\alpha,\beta}(\lambda u) = \lambda \bar{R}_\rho^{\alpha,\beta}(u)$ for $\lambda \geq 0$ and $u \in L^1$. It furthermore obeys the estimate $\alpha TV \leq \bar{R}_\rho^{\alpha,\beta} \leq R_\rho^{\alpha,\beta}$.*

Proof. See [7, Proposition 4.4]. □

3.5 Functionals $\mathbf{TVX}_0$ and $\mathbf{TVX}_1$

The next two examples are the results of using the discrete or geodesic metrics from Examples 3.8 and 3.9 with functional $\bar{R}_\rho^{\alpha,\beta}$. For the derivation see again [7, Examples 4.5, Example 4.6].

Example 3.22 ($\mathrm{TVX}_0^{\alpha,\beta}$)**.** *For $\alpha, \beta > 0$ and ρ_0 the discrete metric from Example 3.8 we can compute*

$$\mathrm{TVX}_0^{\alpha,\beta}(u) := \bar{R}_{\rho_0}^{\alpha,\beta}(u) = \inf_{\substack{\mu \in \mathcal{M}(\Omega \times S^1) \\ (u,\mu) \in M_\nabla}} \alpha \|\mu\|_{\mathcal{M}} + \frac{\beta}{2} \|\nabla_\nu \mu\|_{\mathcal{M}}.$$

Example 3.23 ($\mathrm{TVX}_1^{\alpha,\beta}$)**.** *Let $\alpha, \beta > 0$ and ρ_1 be the geodesic metric from Example 3.9. Then we define for each $\psi \in C^\infty(\Omega \times S^1)$ the projection $(\pi_x \psi)(x) = \int_{S^1} \psi(x, \nu) \, \mathrm{d}\nu$. The set*

$$\widetilde{M}_{\rho_1}(\Omega) = \left\{ (x, \nu) \mapsto \psi(x, \nu) + \varphi(x) \;\middle|\; \psi \in C_c^\infty(\Omega \times S^1), \pi_x \psi = 0, \|\partial_\nu \psi\|_\infty \leq 1, \varphi \in C_c^\infty(\Omega) \right\}$$

can be identified to be sufficient to test with Definition 3.11 in order to obtain T_{ρ_1}. Therefore

$$\mathrm{TVX}_1^{\alpha,\beta}(u) := \bar{R}_{\rho_1}^{\alpha,\beta}(u) = \inf_{\substack{\mu \in \mathcal{M}(\Omega \times S^1) \\ (u,\mu) \in M_\nabla}} \sup_{\substack{\psi \in C_c^\infty(\Omega \times S^1) \\ \pi_x \psi = 0 \\ \|\partial_\nu \psi\|_\infty \leq 1}} \alpha \|\mu\|_{\mathcal{M}} + \beta \langle \nabla_\nu \mu, \psi \rangle.$$

3.6 Applications for imaging problems with L^2 data term

As our intermediate goal was to derive curvature-dependent functionals that can be applied to different image problems, we conclude this section with the final theorem that clarifies the existence of solutions when using convex L^2 data terms.

Theorem 3.24. *Let $G : L^2(\Omega) \to (-\infty, \infty]$ be bounded from below, convex, lower semi-continuous and such that*

$$G(u^n) \to \infty \quad \textit{whenever} \quad \left\{ \left|\int_\Omega u^n \,\mathrm{d}x\right| \to \infty \textit{ and } \left\| u^n - |\Omega|^{-1} \int_\Omega u^n \,\mathrm{d}x \right\|_2 \textit{ is bounded.} \right.$$

Then, for each $\alpha, \beta > 0$ there exists a solution u^* of the variational problem

$$\min_{u \in L^2(\Omega)} \bar{R}_\rho^{\alpha,\beta}(u) + G(u).$$

In case that G is strictly convex, the solution is unique if the minimum is finite.

Proof. See [7, Theorem 4.7]. □

3.7 Data terms

The last chapter will provide numerical examples of the capabilities of the new discretization that will be proposed in the next chapter. Therefore we will introduce a few sample applications. The first ones have mostly been adopted from the original article, and will be used to compare the new discretization with the initial one from [7]. The thin structure preserving noise removal will be used to demonstrate the behavior of the regularization framework in a field where we expect better results than with classical first-order smoothness conditions.

3.7.1 Noise removal

The first application is noise removal. We will consider Gaussian and impulse noise which can be treated using an L^p data fidelity term. For $1 \leq p \leq 2$ and a noisy image $f \in L^p(\Omega)$ we define

$$G(u) = \frac{1}{p} \int_\Omega |u - f|^p \,\mathrm{d}x.$$

For the verification that Theorem 3.24 is applicable compare with [7, Example 4.9]. Notice that for $p = 2$ the functional is strictly convex and hence in that case the solution is unique.

3.7.2 Binary segmentation

As a second example we are going to address binary segmentation, which is the task of separating the domain of an image into two partitions, such as foreground and background. Let $f \in L^1(\Omega)$ be an external segmentation field that is negative for points likely to be in the background and positive for points likely to be in the foreground. The associated data term is then defined as

$$G(u) = \int_\Omega f u \,\mathrm{d}x + \mathbb{1}_D^\infty(u)$$

where $D = \{u \in L^2(\Omega) \mid 0 \leq u(x) \leq 1 \text{ a.e. in } \Omega\}$. The assumptions for Theorem 3.24 are again satisfied, see [7, Example 4.8].

3.7.3 Inpainting

Next we apply the regularization framework to image inpainting. Image inpainting can be used to restore missing information in an image. Consider a non-null set $\Omega' \subset \Omega$ on which the incomplete image $\mathcal{I} \in L^2(\Omega')$ is defined. Then the data term is defined as

$$G(u) = \begin{cases} 0 & \text{if } u|_{\Omega'} = \mathcal{I}, \\ \infty & \text{otherwise.} \end{cases}$$

Compare with [7, Example 4.10] for the existence of a minimizer for Theorem 3.24. However, that minimizer can only be finite if the data has enough regularity.

3.7.4 Deconvolution

As a variation of standard noise removal we consider the task of image deconvolution, where the input image is subject to blurring and (optionally) noise. For $1 \leq p \leq 2$, a blurred and noisy image $f \in L^p(\Omega)$, and a bounded linear operator $\mathrm{PSF} : L^p(\Omega) \to L^p(\Omega)$ satisfying

$$\|\mathrm{PSF}\, u^n\|_p^p \to \infty$$

whenever the sequence (u^n) satisfies

$$\left| \int_\Omega u^n \,\mathrm{d}x \right| \to \infty \text{ and } \left\{ \left\| u^n - |\Omega|^{-1} \int_\Omega u^n \,\mathrm{d}x \right\|_2 \right\} \text{ is bounded,} \tag{3.4}$$

we define the data term for image deconvolution as

$$G(u) = \frac{1}{p} \|\mathrm{PSF}\, u - f\|_p^p .$$

Of course Theorem 3.24 is applicable as G is non-negative, convex and continuous, and satisfies $G(u^n) \to \infty$ if (u^n) satisfies (3.4) since

$$\|\mathrm{PSF}\, u^n - f\|_p^p \geq 2^{1-p} \|\mathrm{PSF}\, u^n\|_p^p - \|f\|_p^p \to \infty.$$

3.7.5 Thin structures preserving noise removal

As a last example we will introduce a form of noise removal that is well suited to preserve thin, line-like structures as, for example, can be found in medical images of blood vessel networks.

The images in consideration consist of connected line segments with the special property that whenever a line branches into multiple segments, the total width of these new lines remains the same as the width of the original one. Moreover, when two lines merge the newly formed segment receives the combined width of the two segments. Instead of considering a 'width', we will more generally use a density to describe these line segments, which can, for instance be measured as the intensity of gray values. Due to the special structure of these images they can be viewed as a (directed) graph, where at every interior node the sum of all incoming edge weights equals the sum of all outgoing edge weights. This property further enables us to find an equivalent level-set representation of the image.

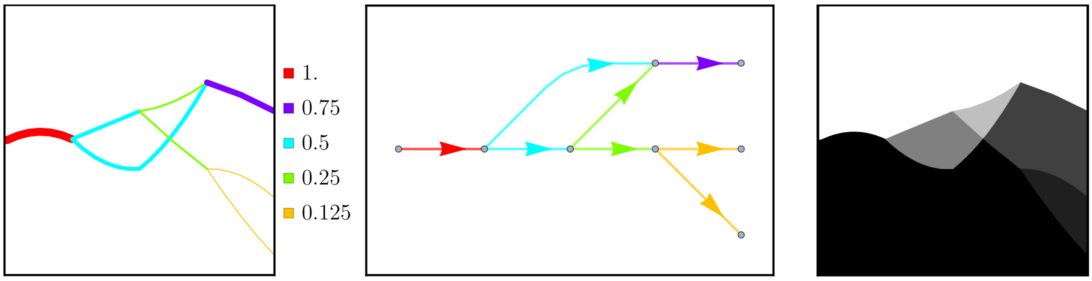


Figure 3.10: *Image with balanced flows, its associated graph and its level set representation.*

In summary, we consider an input density $\mathcal{I} \in \mathcal{M}(\Omega)$ that has been perturbed by noise. We will use the same functionals to apply regularity to the output as before, but the data-fidelity measure will act on the projection of the lifted quantity onto $\mathcal{M}(\Omega)$.

Hence for a data-fidelity measure $G : \mathcal{M}(\Omega) \times \mathcal{M}(\Omega) \to [0, \infty]$ we consider the following optimization problem

$$\inf_{\substack{\mu \in \mathcal{M}(\Omega \times S^1) \\ (u,\mu) \in M_\nabla}} \alpha \|\mu\|_{\mathcal{M}} + \beta T_\rho(\mu) + G(\int_{S^1} \mathrm{d}\mu, \mathcal{I}),$$

where u corresponds to the level set representation and the projection of μ to the noise reduced density, i.e. the quantity of interest. The level set image is merely a byproduct of the computation, and is of no particular interest. Since the objective is also independent of u we will reformulate the problem independently of u. First notice that for any $(u, \mu) \in M_\nabla$ and $\varphi \in C_c^\infty$

$$\int_{\Omega \times S^1} \nabla\varphi(x) \cdot \nu \, \mathrm{d}\mu(x, \nu) = \int_{\Omega \times S^1} (\nabla\varphi(x))^\perp \cdot \nu^\perp \, \mathrm{d}\mu(x, \nu) = -\int_\Omega u \operatorname{div}((\nabla\varphi)^\perp) \, \mathrm{d}x = 0 \tag{3.5}$$

as $\operatorname{div}((\nabla\varphi)^\perp) = \operatorname{curl} \nabla\varphi = 0$. Therefore the modified optimization problem is

$$\inf_{\mu \in M_\Theta} \alpha \|\mu\|_{\mathcal{M}} + \beta T_\rho(\mu) + G(\int_{S^1} \mathrm{d}\mu, \mathcal{I}) \tag{3.6}$$

where the constraints set M_Θ is given by

$$M_\Theta = \left\{ \mu \in \mathcal{M}(\Omega \times S^1) \;\middle|\; \mu \geq 0, \int_{\Omega \times S^1} \nabla\varphi(x) \cdot \nu \, \mathrm{d}\mu(x, \nu) = 0 \text{ for all } \varphi \in C_c^\infty(\Omega, \mathbb{R}) \right\}. \tag{3.7}$$

Proposition 3.25. *The set M_Θ according to* (3.7) *is non-empty, convex and sequentially closed with respect to weak* convergence in $\mathcal{M}(\Omega \times S^1)$.*

Proof. The proof is a slightly modified version of the proof of [7, Proposition 4.1]. M_Θ is of course non-empty and convex. For a sequence $\mu^n \overset{*}{\rightharpoonup} \mu$ in $\mathcal{M}(\Omega \times S^1)$ and for each $\psi \in C_0(\Omega \times S^1)$ with $\psi \geq 0$ it holds that

$$\int_{\Omega \times S^1} \psi \, \mathrm{d}\mu = \lim_{n \to \infty} \int_{\Omega \times S^1} \psi \, \mathrm{d}\mu^n \geq 0$$

since each μ^n is positive, which establishes that μ is a positive measure. The weak* convergence further implies convergence of the integrals

$$\int_{\Omega\times S^1} \nabla\varphi(x)\cdot\nu\,\mathrm{d}\mu(x,\nu) = \lim_{n\to\infty}\int_{\Omega\times S^1}\varphi(x)\cdot\nu\,\mathrm{d}\mu^n(x,\nu) = 0.$$

for each $\varphi\in C_c^\infty$. Consequently M_Θ is weak*-closed. □

Theorem 3.26. *Let $G:\mathcal{M}(\Omega)\to[0,\infty]$ be weak* lower semi-continuous and convex. Then, for each $\alpha>0$ and $\beta>0$ there exists a solution μ^* of the variational problem*

$$\inf_{\mu\in M_\Theta}\alpha\left\|\mu\right\|_{\mathcal{M}}+\beta T_\rho(\mu)+G(\int_{S^1}\mathrm{d}\mu).$$

Proof. The proof is an application of the direct method. If the objective function $F(\mu)=\alpha\left\|\mu\right\|_{\mathcal{M}}+\beta T_\rho(\mu)+G(\int_{S^1}\mathrm{d}\mu)$ is constant ∞ then statement is trivial. Hence we assume that F is finite for at least one $\mu\in M_\Theta$. Since $\left\|\mu\right\|_{\mathcal{M}}, T_\rho(\mu)$ and $G(\mu)$ are all non-negative F is bounded below. Hence there exists a minimizing sequence (μ^n) with $F(\mu^n)\to\inf_{\mu\in M_\Theta}F(\mu)>-\infty$. Without loss of generality we assume that $(F(\mu^n))$ is finite for every $n\geq1$ and monotonically decreasing. Thus (μ^n) is bounded, as

$$\left\|\mu^n\right\|_{\mathcal{M}}\leq\alpha^{-1}(\alpha\left\|\mu^n\right\|_{\mathcal{M}}+\beta T_\rho(\mu^n)+G(\int_{S^1}\mathrm{d}\mu^n))\leq\alpha^{-1}F(\mu^1)<\infty.$$

Consequently there exists a subsequence (μ^{n_k}) and a $\mu^*\in\mathcal{M}(\Omega\times S^1)$ with $\mu^{n_k}\overset{*}{\rightharpoonup}\mu^*$ and from Proposition 3.25 it follows that $\mu^*\in M_\Theta$, since M_Θ is weak* sequentially closed. Moreover, since $\left\|\mu\right\|_{\mathcal{M}}$, $T_\rho(\mu)$ and G are sequentially weak* lower semi-continuous we can conclude

$$F(\mu^*)\leq\liminf_{n\to\infty}F(\mu^n),$$

implying that μ^* is a minimizer. □

Because of the similarity to the initial definition of functionals TVX_0 and TVX_1 we use the following naming convention.

Notation 3.27. *Using the notation from Theorem 3.26 we refer to the problem*

$$\inf_{\mu\in M_\Theta}\alpha\left\|\mu\right\|_{\mathcal{M}}+\frac{\beta}{2}\left\|\nabla_\nu\mu\right\|+G(\int_{S^1}\mathrm{d}\mu)$$

as TVX_0 *regularization of G and*

$$\inf_{\mu\in M_\Theta}\sup_{\substack{\psi\in C_c^\infty(\Omega\times S^1)\\ \int_{S^1}\psi\,\mathrm{d}\nu=0\\ \left\|\partial_\nu\psi\right\|_\infty\leq1}}\alpha\left\|\mu\right\|+\beta\langle\nabla_\nu\mu,\psi\rangle+G(\int_{S^1}\mathrm{d}\mu)$$

as TVX_1 *regularization of G.*

As data fidelity measures on $\mathcal{M}(\Omega)$ we will use the Wasserstein-1 and the H^{-1} metrics as introduced in the first chapter.

Remark 3.28. *Another intention for this example is to study the quality of this type of noise removal in a two-dimensional setting as it could possibly be applied in biological cell tracking in a video sequence.*

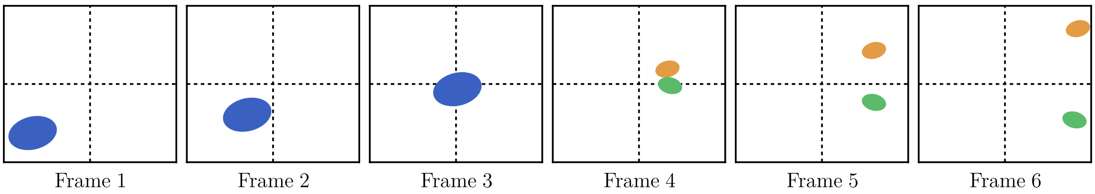


Figure 3.11: *Illustration of a video sequence of cells. In the first three frames a cell moves around and then divides itself into two moving daughter cells.*

A video sequence can be viewed as a three dimensional image where the third dimension corresponds to time, i.e. informally speaking a discrete three-dimensional image can be obtained by stacking the frames one above the other. When applying this to video frames with moving cells as illustrated in Figure 3.11 we obtain three-dimensional paths on which the cells travel. When a cell divides itself the volume (density) of the cell is distributed to a set of new paths originating at the point $(x, t) \in \Omega \times \mathbb{R}_+$ *on which the cell divided itself.*

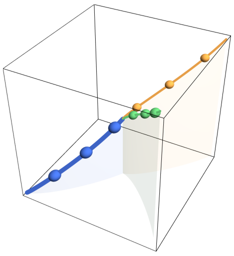


Figure 3.12: *Video of moving cells as three-dimensional image. At infinite frame rate each cell then moves on a continuous path until the cell divides itself. The points along the paths correspond to discrete locations as in Figure 3.11.*

In this sense this thesis considers one-dimensional cell tracking, where each cell

- *has an initial width (density),*
- *can move left and right,*
- *can divide itself into new cells that in sum have the same width (density) as the cell itself.*

In this scenario it is of special interest that the regularization is able to recover missing parts of the cells' path and hence in a sense serves as interpolation.

The generalization of the curvature regularization framework to three-dimensional images is an active field of research, and this thesis will only cover two-dimensional cases.

4 Discretization via Line Measures

The authors of the article ”Convex relaxation of a class of vertex penalizing functionals” [7] proposed a discretization for the two functionals TVX_0 and TVX_1 based on a non-local finite differences scheme, which treats the lifted quantity $\mu \in \mathcal{M}(\Omega \times S^1)$ as a differentiable function. Note that this is not an accurate representation as we expect μ to be composed of line measures with support only on one-dimensional subsets in $\Omega \times S^1$.

This thesis will discuss a new discretization that has been modeled based on representing μ by line measures. The space of directions S^1 will be discretized into K distinct orientations that directly connect nodes in a discrete grid in Ω. Hence we will not require any interpolation between nodal values, which is one of the disadvantages of the initial approach because it introduces dissipation and prevents measures from concentrating on one-dimensional lines.

First we will introduce the discrete operators, conditions and energies, and then present two methods of numerically computing them. One approach utilizes the first-order primal-dual algorithm from [9], and the other employs standard linear or quadratic programming.

4.1 Discrete setting

During the entire section we consider $\Omega = \big(0, Mh\big) \times \big(0, Nh\big)$ to be an open rectangular region that is discretized on a covering, regular Cartesian grid

$$\bar{\Omega}_h = \{(ih, jh) \mid 0 \leq i \leq M, 0 \leq j \leq N\}$$

where M denotes the height, N the width of a discrete image and h the size of a discrete pixel. The grid points are denoted by $x_{ij} = (ih, jh)$ for $(i,j) \in \bar{\Omega}_1$, where $\bar{\Omega}_1$ is simply the set of all indices belonging to grid points inside $\bar{\Omega}_h$, a notation we will frequently use.

Moreover we consider the space of piecewise constant images

$$\mathcal{U}_h = \Big\{U : \Omega \to \mathbb{R} \;\Big|\; U|_{(ih,ih+h)\times(jh,jh+h)} = \text{const}\,,\ \text{for all } 0 \leq i \leq M-1, 0 \leq j \leq N-1\Big\}\,,$$

referred to as the space of *discrete pixel images* and for any image $u : \Omega \to \mathbb{R}$ we set $U : \Omega_h \to \mathbb{R} \in \mathcal{U}_h$ to the L^2 projection of u onto the set $\mathcal{U}_h$. To discretize the lifted quantities $\mu \in \mathcal{M}(\Omega \times S^1)$ we will consider weak-* approximations.

As indicated in the introduction we expect μ to be composed of line measures with support only on one-dimensional subsets in $\Omega \times S^1$. Hence we will provide a new method that, different from the one proposed in [7], will not treat μ as a differentiable functional.

From the set of feasible directions on the unit Cartesian grid

$$\{(\nu_1, \nu_2) \in \mathbb{Z} \times \mathbb{Z} \mid \gcd(\nu_1, \nu_2) = 1\}$$

we select K distinct directions ν_k^* and rescale them to be consistent to our grid with scaling h, i.e. $\nu_k = h\nu_k^*$. This choice ensures that all ν_k are perfectly aligned to the grid, and that we do not select positive multiples of the same direction.

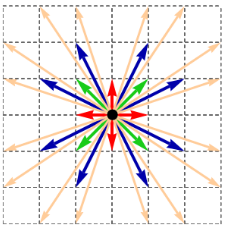


Figure 4.1: *In all computations we used the illustrated choice of directions, where for $K = 4$ we selected the orientations indicated by the red arrows, for $K = 8$ by the red and green arrows and for $K = 16$ additionally by the blue arrows. Occasionally the number of discrete orientations will be set to $K = 32$ utilizing all indicated arrows.*

The lifted quantity μ will then be discretized on a regular grid

$$\bar{\Omega}_h \times S^1_K = \Big\{(ih, jh, \theta_k) \,\Big|\, (i,j) \in \bar{\Omega}_1,\, 1 \le k \le K\Big\}$$

where $\theta_k \in [0, 2\pi)$ is the angle associated with discrete direction ν_k. At every node of the discrete image grid $\bar{\Omega}_h$ we consider K line measure segments with orientations ν_k connecting node x_{ij} with nodes $x_{ij} + \nu_k$. Using this representation we approximate measures $\mu \in \mathcal{M}(\Omega \times S^1)$ by discrete versions

$$\sum_{(i,j)\in\bar{\Omega}_1} \sum_{k=1}^{K} a^k_{ij} \mu_{ijk} \tag{4.1}$$

where $a^k_{ij} \ge 0$ and μ_{ijk} are the line measure segments with orientation ν_k and density 1 at node $x_{ij} \in \bar{\Omega}_h$, i.e. measures defined by

$$\int_{\Omega\times S^1} \varphi(x, \nu)\, \mathrm{d}\mu_{ijk}(x, \nu) = \int_{\{c^k_{ij}\}} \varphi(x, \frac{\nu_k}{|\nu_k|})\, \mathrm{d}\mathcal{H}^1(x) \qquad \forall \varphi \in C_c(\Omega \times S^1, \mathbb{R})$$

with $\{c^k_{ij}\} = \{x_{ij} + t\nu_k \mid t \in [0, 1]\}$ denoting the line segment between nodes x_{ij} and $x_{ij} + \nu_k$. As segments $\{c^k_{ij}\}$ can naturally be parameterized by $t \mapsto x_{ij} + t\frac{\nu_k}{|\nu_k|}$ the line integral resolves to

$$\int_{\{c^k_{ij}\}} \varphi(x, \frac{\nu_k}{|\nu_k|})\, \mathrm{d}\mathcal{H}^1(x) = \int_0^{|\nu_k|} \varphi(x_{ij} + t\frac{\nu_k}{|\nu_k|}, \frac{\nu_k}{|\nu_k|})\, \mathrm{d}t.$$

Remark 4.1. *To keep notation concise we identify nodes in $\bar{\Omega}_h \times S^1_K$ by (x_{ij}, ν_k), i.e. by the direction ν_k rather than the associated angle θ_k.*

Further notice that in general grid points along the vertical axes are not equally spaced as the discrete directions ν_k are not required to be uniformly distributed on the unit circle. The spacing *between node (x_{ij}, ν_{k_1}) and (x_{ij}, ν_{k_2}) can be computed via the geodesic metric ρ_1, i.e.*

$$\operatorname{dist}\left((x_{ij}, \nu_{k_1}), (x_{ij}, \nu_{k_2})\right) = \rho_1(\nu_{k_1}, \nu_{k_2}).$$

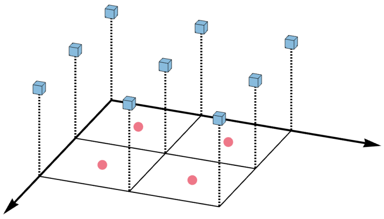


Figure 4.2: *The lower grid displays the standard cell-centered discretization of the pixel image. The discrete lifted quantity however is stored node-centered as it is computed between cells of the image. Further notice that for any illustration we will use the top-left corner as origin of the grid, as this is consistent with the matrix ordering. Moreover the two grids do not share the same size.*

Observe that images and the lifted quantities are discretized on grids with different numbers of nodes. While a discrete image is defined in $\mathbb{R}^{M\times N}$, lifted measures are defined through coefficients in $\mathbb{R}^{(M+1)\times(N+1)\times K}$. Because it is used very frequently we will apply the following shorthand

$$\bar{M} = M + 1, \quad \bar{N} = N + 1$$

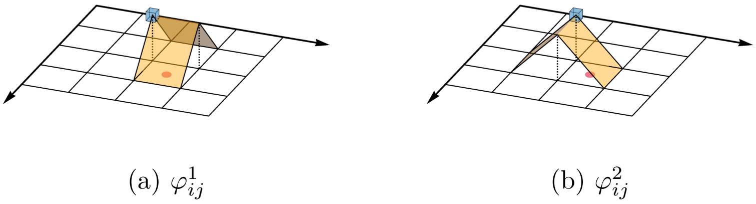
which follows the notion that we cover $\bar{\Omega}$ with $\bar{M}\bar{N}$ points x_{ij}.

4.1.1 Discrete compatibility condition

First we need an approximation of the image gradient ∇u. As the discrete image U only provides constant values inside each cell, the best available approximation of first-order derivatives is based on a first-order finite differences reconstruction. For every $i, j \in \mathbb{N}_0$ consider the following two hat functions:

$$\varphi^1_{ij}(x) = \frac{1}{h^3}\mathbb{1}_{\substack{(ih-h,ih+h)\\ \times(jh,jh+h)}}(x)\begin{pmatrix} h - |x_1 - ih| \\ 0 \end{pmatrix},$$

$$\varphi^2_{ij}(x) = \frac{1}{h^3}\mathbb{1}_{\substack{(ih,ih+h)\\ \times(jh-h,jh+h)}}(x)\begin{pmatrix} 0 \\ h - |x_2 - jh| \end{pmatrix}.$$



(a) φ^1_{ij}

(b) φ^2_{ij}

Figure 4.3: *Test functions $\varphi^{1,2}_{ij}$. In both images the red circle indicates the cell $(ih, ih+h) \times (jh, jh+h)$ and the blue cube the associated node (ih, jh).*

Inserting φ_{ij}^1 into the compatibility condition (3.2) yields, for $1 \leq m \leq M-1$ and $0 \leq n \leq N-1$

$$\begin{aligned}
\int_\Omega \nabla u \cdot \varphi_{ij}^1 \,\mathrm{d}x &= \frac{1}{h^3} \int_{\substack{[ih-h,ih+h]\\ \times[jh,jh+h]}} \frac{\partial u}{\partial x_1}(h - |x_1 - ih|)\,\mathrm{d}x \\
&= \frac{1}{h^3} \int_{\substack{[ih-h,ih]\\ \times[jh,jh+h]}} \frac{\partial u}{\partial x_1}(h + x_1 - ih)\,\mathrm{d}x + \frac{1}{h^3} \int_{\substack{[ih,ih+h]\\ \times[jh,jh+h]}} \frac{\partial u}{\partial x_1}(h - x_1 + ih)\,\mathrm{d}x \\
&= \frac{1}{h}\Big(\frac{1}{h^2} \int_{\substack{[ih,ih+h]\\ \times[jh,jh+h]}} u\,\mathrm{d}x - \frac{1}{h^2} \int_{\substack{[ih-h,ih]\\ \times[jh,jh+h]}} u\,\mathrm{d}x\Big) \\
&= \frac{U_{ij} - U_{i-1,j}}{h},
\end{aligned}$$

where U_{ij} by definition corresponds to the pixel value of pixel (i,j). Through the same argument we conclude that $\int_\Omega \nabla u \cdot \varphi_{ij}^2 \,\mathrm{d}x = \frac{U_{ij} - U_{i,j-1}}{h}$ for $0 \leq i \leq M-1, 1 \leq j \leq N-1$. Therefore the first part of the condition reduces to a simple forward difference quotient with between pixel values with Neumann boundary conditions.

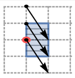
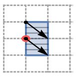
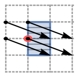
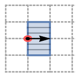
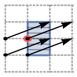
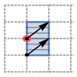
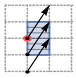
For the second part we insert the discrete form of μ (4.1) into the compatibility condition and test against hat-functions $\varphi_{ij}^{1,2}$, yielding

$$\begin{aligned}
\int_{\Omega\times S^1} \varphi_{ij}^{1,2}(x) \cdot \nu^\perp \,\mathrm{d}\mu &\approx \sum_{(m,n)\in\bar{\Omega}_1} \sum_{k=1}^{K} a_{mn}^k \int_{\Omega\times S^1} \varphi_{ij}^{1,2} \cdot \nu^\perp \,\mathrm{d}\mu_{ijk} \\
&= \sum_{(m,n)\in\bar{\Omega}_1} \sum_{k=1}^{K} a_{mn}^k \int_{\{c_{mn}^k\}} \varphi_{ij}^{1,2} \cdot \frac{\nu_k^\perp}{|\nu_k|} \,\mathrm{d}\mathcal{H}^1(x).
\end{aligned}$$

Because the sets $\{c_{mn}^k\}$ are only simple line segments and $\varphi_{ij}^{1,2}$ has support on open rectangular regions this expression can easily be computed through one of the following cases:

1. The direction ν_k of the line segment is oriented normal to the level sets of $\varphi_{ij}^{1,2}$.
   (i) $\nu_{k,1} = 0 \quad \Rightarrow \quad \int_{\{c_{mn}^k\}} \varphi_{ij}^2 \cdot \nu_k^\perp \,\mathrm{d}\mathcal{H}^1(x) = 0,$
   (ii) $\nu_{k,2} = 0 \quad \Rightarrow \quad \int_{\{c_{mn}^k\}} \varphi_{ij}^1 \cdot \nu_k^\perp \,\mathrm{d}\mathcal{H}^1(x) = 0.$
2. The line segment $\{c_{mn}^k\}$ does not have an intersection with the support of $\varphi_{ij}^{1,2}$.
   (iii) $\{c_{mn}^k\} \cap (ih-h, ih+h) \times (jh, jh+h) = \emptyset \quad \Rightarrow \quad \int_{\{c_{mn}^k\}} \varphi_{ij}^1 \cdot \nu_k^\perp \,\mathrm{d}\mathcal{H}^1(x) = 0,$
   (iv) $\{c_{mn}^k\} \cap (ih, ih+h) \times (jh-h, jh+h) = \emptyset \quad \Rightarrow \quad \int_{\{c_{mn}^k\}} \varphi_{ij}^2 \cdot \nu_k^\perp \,\mathrm{d}\mathcal{H}^1(x) = 0.$
3. The line segment $\{c_{mn}^k\}$ has an intersection with the support of $\varphi_{ij}^{1,2}$, then the intersection itself again is a line segment and the integral evaluations to a non-zero coefficient.

In the last case the integration breaks down to computing the intersection points of the line segment with the rectangular support of $\varphi_{ij}^{1,2}$, and then integrating a polynomial of degree 1. This can easily be precomputed by hand, or using symbolic integration on a computer, as we did. The most important aspect, however, is that in light of the first two observations the compatibility condition becomes sparse. Figure 4.4 exemplary shows a list of precomputed values for φ^1_{ij} and $K = 16$. As φ^2_{ij} can be obtained by clockwise rotation, the corresponding coefficients are essentially the same, except for a negative sign due to the definition of the rotation operator.

<table>
<tr><td></td><td colspan="3">
$k=2$</td><td colspan="2">
$k=3$</td><td colspan="4">
$k=4$</td><td>
$k=5$</td><td colspan="4">
$k=6$</td><td colspan="2">
$k=7$</td><td colspan="3">
$k=8$</td></tr>
<tr><td>m</td><td>$i-2$</td><td>$i-1$</td><td>i</td><td>$i-1$</td><td>i</td><td>$i-1$</td><td>$i-1$</td><td>i</td><td>i</td><td>i</td><td>i</td><td>i</td><td>$i+1$</td><td>$i+1$</td><td>i</td><td>$i+1$</td><td>i</td><td>$i+1$</td><td>$i+2$</td></tr>
<tr><td>n</td><td>j</td><td>j</td><td>j</td><td>j</td><td>j</td><td>$j-1$</td><td>j</td><td>$j-1$</td><td>j</td><td>j</td><td>$j-1$</td><td>j</td><td>$j-1$</td><td>j</td><td>j</td><td>j</td><td>j</td><td>j</td><td>j</td></tr>
<tr><td>$\frac{1}{|\nu_k|}\int_{\{c^k_{mn}\}} \varphi^1_{ij} \cdot \nu^\perp_k \,\mathrm{d}\mathcal{H}^1(x)$</td><td>$-\frac{1}{4}$</td><td>$-\frac{1}{2}$</td><td>$-\frac{1}{4}$</td><td>$-\frac{1}{2}$</td><td>$-\frac{1}{2}$</td><td>$-\frac{3}{4}$</td><td>$-\frac{1}{4}$</td><td>$-\frac{1}{4}$</td><td>$-\frac{3}{4}$</td><td>$-1$</td><td>$-\frac{1}{4}$</td><td>$-\frac{3}{4}$</td><td>$-\frac{3}{4}$</td><td>$-\frac{1}{4}$</td><td>$-\frac{1}{2}$</td><td>$-\frac{1}{2}$</td><td>$-\frac{1}{4}$</td><td>$-\frac{1}{2}$</td><td>$-\frac{1}{4}$</td></tr>
</table>

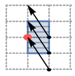
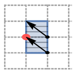
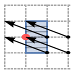
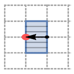
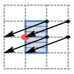
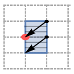
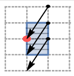
<table>
<tr><td></td><td colspan="3">
$k=10$</td><td colspan="2">
$k=11$</td><td colspan="4">
$k=12$</td><td>
$k=13$</td><td colspan="4">
$k=14$</td><td colspan="2">
$k=15$</td><td colspan="3">
$k=16$</td></tr>
<tr><td>m</td><td>i</td><td>$i+1$</td><td>$i+2$</td><td>i</td><td>$i+1$</td><td>i</td><td>i</td><td>$i+1$</td><td>$i+1$</td><td>i</td><td>$i-1$</td><td>$i-1$</td><td>i</td><td>i</td><td>$i-1$</td><td>i</td><td>$i-2$</td><td>$i-1$</td><td>i</td></tr>
<tr><td>n</td><td>$j+1$</td><td>$j+1$</td><td>$j+1$</td><td>$j+1$</td><td>$j+1$</td><td>$j+1$</td><td>$j+2$</td><td>$j+1$</td><td>$j+2$</td><td>$j+1$</td><td>$j+1$</td><td>$j+2$</td><td>$j+1$</td><td>$j+2$</td><td>$j+1$</td><td>$j+1$</td><td>$j+1$</td><td>$j+1$</td><td>$j+1$</td></tr>
<tr><td>$\frac{1}{|\nu_k|}\int_{\{c^k_{mn}\}} \varphi^1_{ij} \cdot \nu^\perp_k \,\mathrm{d}\mathcal{H}^1(x)$</td><td>$\frac{1}{4}$</td><td>$\frac{1}{2}$</td><td>$\frac{1}{4}$</td><td>$\frac{1}{2}$</td><td>$\frac{1}{2}$</td><td>$\frac{3}{4}$</td><td>$\frac{1}{4}$</td><td>$\frac{1}{4}$</td><td>$\frac{3}{4}$</td><td>$1$</td><td>$\frac{1}{4}$</td><td>$\frac{3}{4}$</td><td>$\frac{3}{4}$</td><td>$\frac{1}{4}$</td><td>$\frac{1}{2}$</td><td>$\frac{1}{2}$</td><td>$\frac{1}{4}$</td><td>$\frac{1}{2}$</td><td>$\frac{1}{4}$</td></tr>
</table>

Figure 4.4: *Precomputed offsets and coefficients used in compatibility condition for a test function φ^1_{ij}. The red circle indicates a node (ih, jh), and the blue rectangle the support of the test function. For every discrete orientation ν_k all nodes are marked where the integration in the compatibility-condition is non-zero. Notice, that the pixel spacing has been set to $h = 1$.*

Combining the previous statements we obtain the *discrete compatibility condition*

$$\sum_{(m,n)\in\bar{\Omega}_1} \sum_{k=1}^{K} a^k_{mn} \int_{\{c^k_{mn}\}} \varphi^1_{ij} \cdot \frac{\nu^\perp_k}{|\nu_k|} \,\mathrm{d}\mathcal{H}^1(x) = \frac{U_{ij} - U_{i-1,j}}{h}, \tag{4.2}$$

$$\sum_{(m,n)\in\bar{\Omega}_1} \sum_{k=1}^{K} a^k_{mn} \int_{\{c^k_{mn}\}} \varphi^2_{ij} \cdot \frac{\nu^\perp_k}{|\nu_k|} \,\mathrm{d}\mathcal{H}^1(x) = \frac{U_{ij} - U_{i,j-1}}{h}, \tag{4.3}$$

where due to boundary conditions

- (4.2) must be satisfied for $1 \leq i \leq M-1$ and $0 \leq j \leq N-1$,
- (4.3) must be satisfied for $0 \leq i \leq M-1$ and $1 \leq j \leq N-1$.

4.1.2 Discrete directional derivative

The next important step is the discretization of the directional derivate $\nabla_\nu \mu$. To compute this we consider a continuous function $\varphi \in \{f \in C_0(\Omega \times S^1, \mathbb{R}) \mid \nabla_x f \in C_0(\Omega \times S^1, \mathbb{R}^2)\}$ and test μ against $(x, \nu) \mapsto \nabla_x \varphi(x, \nu) \cdot \nu$. Then for each line measure μ_{ijk} we have

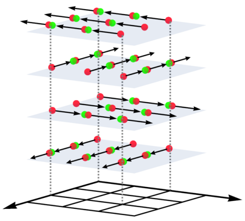
$$\begin{aligned}\int_{\Omega\times S^1} \nabla_x \varphi(x,\nu)\cdot\nu \,\mathrm{d}\mu_{ijk} = & \int_{\{c^k_{ij}\}} \nabla_x \varphi(x, \frac{\nu_k}{|\nu_k|}) \cdot \frac{\nu_k}{|\nu_k|} \,\mathrm{d}x \\ = & \int_0^{|\nu_k|} \frac{\partial}{\partial t} \varphi(x_{ij} + t\frac{\nu_k}{|\nu_k|}, \frac{\nu_k}{|\nu_k|}) \,\mathrm{d}t \\ = & \varphi^k_{ij+\nu^*_k} - \varphi^k_{ij},\end{aligned}$$

where $\varphi_{ij}^k = \varphi(x_{ij}, \frac{\nu_k}{|\nu_k|})$ and $\varphi_{ij+\nu_k^*}^k = 0$ if $(i,j) + \nu_k^* \notin \bar{\Omega}_1$ due to φ being in $C_c(\Omega \times S^1)$. With this representation the discrete form of $\nabla_\nu \mu$ is computed as follows

$$\begin{aligned}
-\langle \nabla_\nu \mu, \varphi \rangle = \int_{\Omega \times S^1} \nabla_x \varphi(x,\nu) \cdot \nu \, \mathrm{d}\mu &\approx \sum_{(i,j)\in\bar{\Omega}_1} \sum_{k=1}^{K} a_{ij}^k \int_{\Omega \times S^1} \nabla_x \varphi(x, \frac{\nu_k}{|\nu_k|}) \cdot \nu_k \, \mathrm{d}\mu_{ijk} \\
&= \sum_{(i,j)\in\bar{\Omega}_1} \sum_{k=1}^{K} a_{ij}^k (\varphi_{ij+\nu_k^*}^k - \varphi_{ij}^k) \\
&= \sum_{k=1}^{K} \sum_{\substack{(i,j)\in\bar{\Omega}_1 \\ (i,j)-\nu_k^*\in\bar{\Omega}_1}} \varphi_{ij}^k a_{ij-\nu_k^*}^k - \sum_{k=1}^{K} \sum_{(i,j)\in\bar{\Omega}_1} \varphi_{ij}^k a_{ij}^k,
\end{aligned} \tag{4.4}$$

where $a_{ij-\nu_k^*}^k$ denotes the linear coefficient at line measure $\mu_{i-\nu_{k,1}^*, j-\nu_{k,2}^*, k}$. The weight $a_{ij-\nu_k^*}^k$ then belongs to the incoming, and a_{ij}^k to the outgoing line measure segment at node x_{ij} with orientation ν_k.

Figure 4.5: *Illustration of flows for the directional derivative $\nabla_\nu \mu$ for $K = 4$. The red circles indicate outgoing flow, the green circles incoming flow. As line measures for different ν_k are lifted to varying heights every node has only one incoming and one outgoing flow.*



4.1.3 Discrete partial derivative in label direction

To implement TVX$_1$ we additionally need to deal with the constrained optimization problem

$$\sup_{\substack{\varphi\in C_c^\infty(\Omega\times S^1) \\ \int_{S^1} \varphi(x,\nu)\,\mathrm{d}\nu = 0 \\ \|\partial_\nu \varphi\|_\infty \le 1}} \langle \nabla_\nu \mu, \varphi \rangle.$$

Using (4.4) the expression evaluates to

$$\sup_{\substack{\varphi\in C_c^\infty(\Omega\times S^1) \\ \int_{S^1} \varphi(x,\nu)\,\mathrm{d}\nu = 0 \\ \|\partial_\nu \varphi\|_\infty \le 1}} \sum_{k=1}^{K} \sum_{\substack{(i,j)\in\bar{\Omega}_1 \\ (i,j)-\nu_k^*\in\bar{\Omega}_1}} \varphi_{ij}^k a_{ij-\nu_k^*}^k - \sum_{k=1}^{K} \sum_{(i,j)\in\bar{\Omega}_1} \varphi_{ij}^k a_{ij}^k$$

where $\varphi_{ij}^k = \varphi(x_{ij}, \frac{\nu_k}{|\nu_k|})$ is only evaluated at discrete point values. Therefore the constraints may be reformulated as

$$\sup_{\substack{\sum_{k=1}^K \varphi_{ij}^k = 0 \\ |\varphi_{ij}^k - \varphi_{ij}^{k-1}| \le \rho_1(\nu_k, \nu_{k-1})}} \sum_{k=1}^{K} \sum_{\substack{(i,j)\in\bar{\Omega}_1 \\ (i,j)-\nu_k^*\in\bar{\Omega}_1}} \varphi_{ij}^k a_{ij-\nu_k^*}^k - \sum_{k=1}^{K} \sum_{(i,j)\in\bar{\Omega}_1} \varphi_{ij}^k a_{ij}^k \tag{4.5}$$

with ρ_1 denoting the geodesic distance in S^1 and periodic boundary conditions, i.e. $\nu_0 = \nu_K$ and $\varphi_{ij}^0 = \varphi_{ij}^K$.

4.1.4 Discrete flow balancing condition

In the specialized noise removal the compatibility condition is replaced by flow-balancing condition (3.5). For every line measure μ_{ijk} and $\varphi \in C_0^\infty(\Omega)$ we have as usual

$$\int_{\Omega\times S^1} \nabla\varphi(x)\cdot\nu\,\mathrm{d}\mu_{ijk} = \int_{\{c_{ij}^k\}} \nabla\varphi(x)\cdot\frac{\nu_k}{|\nu_k|}\,\mathrm{d}\mathcal{H}^1(x) = \int_0^{|\nu_k|}\frac{\partial}{\partial t}\varphi(x+t\frac{\nu_k}{|\nu_k|})\,\mathrm{d}t = \varphi_{ij+\nu_k}-\varphi_{ij}$$

where $\varphi_{ij} = \varphi(x_{ij})$ and $\varphi_{ij+\nu_k^*} = 0$ whenever $(i,j)+\nu_k^* \notin \bar{\Omega}_1$. Hence

$$\begin{aligned}
\int_{\Omega\times S^1} \nabla\varphi(x)\cdot\nu\,\mathrm{d}\mu(x,\nu) &\approx \sum_{(i,j)\in\Omega_1}\sum_{k=1}^{K} a_{ij}^k \int_{\{c_{ij}^k\}} \nabla\varphi(x)\cdot\frac{\nu_k}{|\nu_k|}\,\mathrm{d}\mathcal{H}^1(x)\\
&= \sum_{(i,j)\in\Omega_1}\sum_{k=1}^{K} a_{ij}^k(\varphi_{ij+\nu_k}-\varphi_{ij})\\
&= \sum_{(i,j)\in\Omega_1}\varphi_{ij}\sum_{k=1}^{K}(a_{ij-\nu_k^*}^k - a_{ij}^k) = 0 \text{ for all } \varphi\in C_0^\infty
\end{aligned}$$

with $a_{ij-\nu_k^*}^k = 0$ whenever $(i,j)-\nu_k^* \notin \Omega_1$, which is satisfied if

$$\sum_{\substack{1\le k\le K\\ (i,j)-\nu_k^*\in\bar{\Omega}_1}} a_{ij-\nu_k^*}^k - \sum_{k=1}^{K} a_{ij}^k = 0 \qquad \text{for all } (i,j)\in\bar{\Omega}_1 \tag{4.6}$$

The last equation illustrates the name flow-balancing condition, since it ensures that at every node (i,j) the total incoming mass and total outgoing mass of the connected line measures are in equilibrium. Notice that, depending on the directions ν_k, we possibly introduce a small numerical error at the boundary, since the node for the incoming measure might not be discretized.

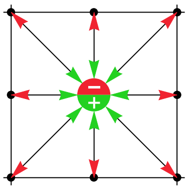


Figure 4.6: *Flow-balancing condition for one node (i,j) and $K = 8$ outgoing and incoming line measures.*

4.1.5 Discrete projection

We also need a discrete projection that maps a measure $\mu \in \mathcal{M}(\Omega\times S^1)$ to a two-dimensional measure $\pi\mu \in \mathcal{M}(\Omega)$,

$$\pi\mu = \int_{S^1}\mathrm{d}\mu.$$

The projection should be properly aligned to the discrete image grid, therefore we have to distribute the point mass stored at cell nodes to the correct cells. In case of vertical or horizontal directions, we split the mass equally to the cells normal to the direction, and in any other case the mass is distributed equally to all cells through which the direction passes.

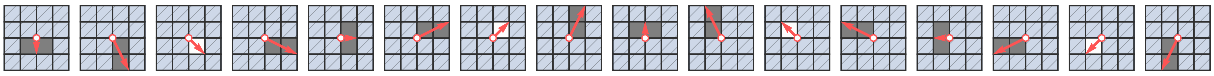


Figure 4.7: *Distribution of mass along the different directions ν_k.*

We define $\mathcal{I}_{ij}$ as the set of nodes

$$\mathcal{I}_{ij} = \Big\{(m,n,k) \;\Big|\; \mathcal{H}^1(\{c_{mn}^k\} \cap [ih, ih+h] \times [jh, jh+h]) > 0\Big\}$$

for which the line measures μ_{mnk} pass through cell (i,j). For every feasible discrete direction ν_k the mass is distributed equally to the adjacent cells, therefore we compute weights ω_k according to

$$\omega_k = \frac{1}{l_k^+} \min\left(1, \max(l_k^-, \frac{1}{2})\right)$$

with $l_k^+ = \max(|\nu_{k,1}^*|, |\nu_{k,2}^*|)$, $l_k^- = \min(|\nu_{k,1}^*|, |\nu_{k,2}^*|)$. The discrete projection is then defined as

$$(\pi\mu)_{ij} = \sum_{(m,n)\in\bar{\Omega}_1} \sum_{k=1}^{K} a_{m,n}^k \omega_k \mathbb{1}_{\mathcal{I}_{ij}}(m,n,k). \tag{4.7}$$

4.2 Matrix representation of discrete operators

All discrete operators defined in the previous section can be formulated in terms of sparse matrices. In order to use standard matrix algebra we implicitly reshape the discrete image into a vector in $\mathbb{R}^{MN}$. In the same way the linear coefficients $a \in \mathbb{R}^{\bar{M}\times\bar{N}\times K}$ of the discrete lifted measure are implicitly reshaped into a vector in $\mathbb{R}^{\bar{M}\bar{N}K}$. But to keep notation simple we use the same symbols interchangeably.

Remark 4.2. *In our computations we always used a linear indexing scheme based on concatenating all columns of the image matrix into one long column vector.*

Example 4.3. *The matrix indices of a discrete image $U \in \mathbb{R}^{3\times4}$ are linearized as follows:*

$$U = \begin{pmatrix} \boxed{1}\, u_{11} & \boxed{4}\, u_{12} & \boxed{7}\, u_{13} & \boxed{10}\, u_{14} \\ \boxed{2}\, u_{21} & \boxed{5}\, u_{22} & \boxed{8}\, u_{23} & \boxed{11}\, u_{24} \\ \boxed{3}\, u_{31} & \boxed{6}\, u_{32} & \boxed{9}\, u_{33} & \boxed{12}\, u_{34} \end{pmatrix}.$$

In the same way the indices of $a \in \mathbb{R}^{3\times4\times3}$ are linearized by concatenating the matrices for each k into one matrix and then using the same linear indexing scheme as with the image U.

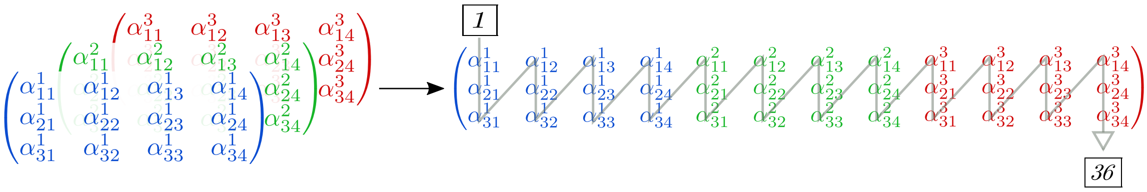


Figure 4.8: *Linearized indexing scheme for three-dimensional coefficients.*

We will use the same naming convention for the matrix operators as was introduced in the original article [7], as most of the results still apply with minor changes. Analogous to the previous section we begin with the discrete compatibility condition.

Let $A \in \mathbb{R}^{2MN \times MN}$ be the forward difference operator defined by

$$(A^1 u)_{ij} = \begin{cases} \frac{u_{ij} - u_{i-1,j}}{h} & \text{if } 1 \le i \le M-1, 0 \le j \le N-1, \\ 0 & \text{otherwise,} \end{cases}$$
$$(A^2 u)_{ij} = \begin{cases} \frac{u_{ij} - u_{i,j-1}}{h} & \text{if } 0 \le i \le M-1, 1 \le j \le N-1, \\ 0 & \text{otherwise.} \end{cases} \tag{4.8}$$

Further denoted by $D \in \mathbb{R}^{2MN \times \bar{M}\bar{N}K}$ is the operator implementing (4.2)

$$(D^1 a)_{ij} = \begin{cases} \sum_{(m,n)\in\Omega_1} \sum_{k=1}^{K} a_{mn}^k \frac{1}{|\nu_k|} \int_{\{c_{mn}^k\}} \varphi_{ij}^1 \cdot \nu_k^\perp \, \mathrm{d}\mathcal{H}^1(x) & \text{if } 1 \le i \le M-1,\ 0 \le j \le N-1, \\ 0 & \text{otherwise,} \end{cases}$$
$$(D^2 a)_{ij} = \begin{cases} \sum_{(m,n)\in\Omega_1} \sum_{k=1}^{K} a_{mn}^k \frac{1}{|\nu_k|} \int_{\{c_{mn}^k\}} \varphi_{ij}^2 \cdot \nu_k^\perp \, \mathrm{d}\mathcal{H}^1(x) & \text{if } 0 \le i \le M-1,\ 1 \le j \le N-1, \\ 0 & \text{otherwise.} \end{cases}$$

Notice again that, even though the large summation might suggest otherwise, D is sparse, and that all coefficients $\frac{1}{|\nu_k|} \int_{\{c_{mn}^k\}} \varphi_{ij}^{1,2} \cdot \nu_k^\perp \, \mathrm{d}\mathcal{H}^1(x)$ are fully determined.

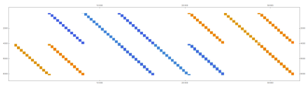


Figure 4.9: *Sparse pattern of D for $K = 8$ directions. The four empty block matrices correspond the directions normal to level sets of φ_{ij}^1 and φ_{ij}^2 respectively.*

Next let $B \in \mathbb{R}^{\bar{M}\bar{N}K \times \bar{M}\bar{N}K}$ be the operator that implements the discrete directional derivative $\nabla_\nu \mu$ according to (4.4)

$$(Ba)_{ijk} = \begin{cases} a_{ij}^k - a_{ij-\nu_k^*}^k & \text{if } (i,j) - \nu_k^* \in \bar{\Omega}_1, \\ a_{ij}^k & \text{otherwise.} \end{cases}$$

Denoted by $C \in \mathbb{R}^{\bar{M}\bar{N}K \times \bar{M}\bar{N}K}$ is the operator that computes the partial derivative ∂_ν according to (4.5)

$$(C\varphi)_{ijk} = (\varphi_{ijk} - \varphi_{ij,k-1})\rho_1(\nu_k, \nu_{k-1})^{-1}$$

utilizing periodic boundary conditions, i.e. $\nu_0 = \nu_K$ and $\varphi_{ij0} = \varphi_{ijK}$. As before $\rho_1 : S^1 \times S^1 \to \mathbb{R}$ denotes the geodesic metric.

Moreover denoted by $P \in \mathbb{R}^{MN \times \bar{M}\bar{N}K}$ is the projection operator (4.7)

$$(Pa)_{ij} = \sum_{(m,n)\in\bar{\Omega}_1} \sum_{k=1}^{K} a_{m,n}^k \omega_k \mathbb{1}_{\mathcal{I}_{ij}}(m,n,k) \, .$$

Notice that we project the measure onto the slightly smaller grid of pixel images.

Finally denoted by $F \in \mathbb{R}^{\bar{M}\bar{N}\times\bar{M}\bar{N}K}$ is the operator implementing the flow-balancing condition (4.6)

$$(Fa)_{ij} = \sum_{\substack{1\le k\le K\\(i,j)-\nu_k^*\in\bar{\Omega}_1}} a^k_{ij-\nu_k^*} - \sum_{k=1}^{K} a^k_{ij}.$$

4.3 Discrete formulations for TVX

The last section showed that all operators required to discretize the proposed regularization framework are linear operators, and therefore a wide range of algorithms is applicable to numerically solve problems of the form 3.24.

In the following we will first make use of the primal-dual algorithm that was also discussed in the original article, and secondly introduce a standard linear programming approach. The two methods require the optimization problem to be posed in a specific form, which we will now formulate using the discrete operators from the previous section.

The following will show that functionals TVX_0 and TVX_1 can be written as linear optimization problems. Since linear programming is a special case of quadratic programming we will later generalize the formulation to allow a larger class of data terms, such as the removal of Gaussian noise based on an ℓ^2 data term.

4.3.1 Discrete TVX_0

The functional TVX_0 is defined as

$$\inf_{\substack{\mu\in M(\Omega\times S^1)\\(u,\mu)\in M_\nabla}} \alpha\,\|\mu\|_{\mathcal{M}} + \frac{\beta}{2}\,\|\nabla_\nu\,\mu\|_{\mathcal{M}}\,.$$

In its discrete form the constraint M_∇ is defined as

$$M_\nabla = \left\{(u,a)\;\middle|\; a^k_{ij}\ge 0, Au = Da\right\}.$$

To apply the primal-dual algorithm from [9] we need a saddle-point formulation. Therefore we introduce Lagrange multipliers $\varphi\in\mathbb{R}^{2MN}$ for the linear constraint $Au = Da$, which yields the desired form:

$$\begin{aligned}\min_{u,a}\max_{\psi,\varphi}\quad & \alpha\sum_{ijk} a^k_{ij}\,|\nu_k| + \beta\langle Ba,\psi\rangle + \langle Au - Da,\varphi\rangle\\ \text{subject to}\quad & a^k_{ij}\ge 0,\\ & |\psi_{ijk}|\le \tfrac{1}{2}.\end{aligned} \tag{4.9}$$

To apply linear programming we need a pure minimization formulation with a linear objective function and linear constraints. We consider the (not yet linear) minimization problem

$$\begin{aligned} \min_{u,a} \quad & \alpha \sum_{ijk} a_{ij}^k \, |\nu_k| + \frac{\beta}{2} \, \|Ba\|_1 \\ \text{subject to} \quad & a \geq 0, \\ & Au = Da. \end{aligned} \tag{4.10}$$

This is of course not the final form, since the objective is still non-linear due to the 1-norm. The next section will show multiple ways for transforming the above problem into a linear program. Since that choice has shown to have a strong impact on the numerical computations and highly depends on the solver, we decided to leave it open at this point.

4.3.2 Discrete TVX_1

The functional TVX_1 is defined as

$$\inf_{\substack{\mu \in \mathcal{M}(\Omega \times S^1) \\ (u,\mu) \in M_\nabla}} \sup_{\substack{\psi \in C_c^\infty(\Omega \times S^1) \\ \int_{S^1} \psi(x,\nu) \, \mathrm{d}\nu = 0 \\ \|\partial_\nu \psi\|_\infty \leq 1}} \alpha \, \|\mu\|_{\mathcal{M}} + \beta \langle \nabla_\nu \, \mu, \psi \rangle .$$

Of course, just like for TVX_0, the discrete set M_∇ is defined as

$$M_\nabla = \left\{ (u,a) \;\middle|\; a_{ij}^k \geq 0, Au = Da \right\} .$$

We again introduce a set of Lagrange multipliers $\varphi \in \mathbb{R}^{2MN}$ to account for constraint $Au = Da$. Further we introduce an auxiliary variable $\zeta \in \mathbb{R}^{\bar{M}\bar{N}K}$ and another set of Lagrange multipliers $v \in \mathbb{R}^{\bar{M}\bar{N}K}$ to handle the constrained optimization (4.5). Then the saddle-point formulation is:

$$\begin{aligned} \min_{u,a,v} \max_{\psi,\varphi,\zeta} \quad & \alpha \sum_{ijk} a_{ij}^k \, |\nu_k| + \beta \langle Ba, \psi \rangle + \langle Au - Da, \varphi \rangle + \langle C\psi - \zeta, v \rangle \\ \text{subject to} \quad & a_{ij}^k \geq 0, \\ & \sum_k \psi_{ijk} = 0, \\ & |\psi_{ijk}| \leq 1. \end{aligned} \tag{4.11}$$

To apply linear programming the system needs to be reformulated as minimization problem. As previously noted the constraint $\int_{S^1} \psi(x,\nu) \, \mathrm{d}\nu = 0$ is optional and serves to increase the numerical stability by reducing the space of allowed test functions. While this has proven to be true for the first-order primal-dual algorithm our tests have shown that the constraint is unnecessary for linear or quadratic programming, and in many cases reduces the rate of convergence. Hence we will drop it here. To rewrite TVX_1 as a minimization problem we can apply proposition 2.26 to the constrained maximization problem (4.5)

$$\sup_{\substack{\psi\in C_c^\infty(\Omega\times S^1)\\ \|\partial_\nu\psi\|_\infty\le 1}} \langle \nabla_\nu\, \mu, \psi\rangle = \sup_{\substack{\psi\in\mathbb{R}^{\bar M\bar N K}\\ \|C\psi\|_\infty\le 1}} (Ba)\cdot\psi = \min_{\substack{\psi\in\mathbb{R}^{\bar M\bar N K}\\ C^T\psi+Ba=0}} \|\psi\|_1\,.$$

Hence with auxiliary variable $p\in\mathbb{R}^{\bar M\bar N K}$ the functional TVX$_1$ can be reformulated as the following (not yet linear) minimization problem

$$\begin{aligned} \min_{u,a,s,p} \quad & \alpha\sum_{ijk} a_{ij}^k\,|\nu_k| + \beta\,\|p\|_1 \\ \text{subject to} \quad & a_{ij}^k \ge 0, \\ & Au = Da, \\ & C^T p + Ba = 0. \end{aligned} \tag{4.12}$$

Again as with TVX$_0$ we did not replace the 1-norm at this point, and thus the formulation is not yet a linear program. This step will be done in the next section, acknowledging that the choice of an equivalent formulation potentially has a strong impact on the numerical computations.

5 Numerical algorithms

With the saddle-point and the minimization formulations available we can utilize different numerical algorithms to compute image problems regularized by TVX_0 or TVX_1 as in Theorem 3.24. First we will provide exact formulations for the primal-dual algorithm and then for a linear programming approach.

5.1 Primal-Dual algorithm

The first-order primal-dual algorithm from [9] can solve minimization problems with known saddle-point structure of the form

$$\min_{x \in X} \max_{y \in Y} \langle \mathcal{K}x, y \rangle + \mathcal{G}(x) - \mathcal{F}^*(y)$$

where X and Y are finite-dimensional vector spaces, $\mathcal{K}$ is a linear operator and $\mathcal{G}$ and $\mathcal{F}^*$ are proper, convex, and lower-semi-continuous functions. We will use the same method as described in [7], which utilizes diagonal preconditioning and an additional overrelaxation step that has been shown to speed up convergence, see [11].

For a linear operator $\mathcal{K} \in \mathbb{R}^{m \times n}$ the diagonal preconditioning matrices $\mathcal{T} \in \mathbb{R}^{n \times n}$, $\mathcal{S} \in \mathbb{R}^{m \times m}$ are defined by

$$\mathcal{T}_{bb} = \Big(\sum_{a=1}^{m} |\mathcal{K}_{ab}| \Big)^{-1}, \qquad \mathcal{S}_{aa} = \Big(\sum_{b=1}^{n} |K_{ab}| \Big)^{-1}. \tag{5.1}$$

To utilize this algorithm it is important that the proximity operators of $\mathcal{G}$ and $\mathcal{F}^*$ have closed-form solutions that can be computed efficiently. This is the case for many standard image problems, as the examples will show. The algorithm is defined as follows.

Algorithm 1 Primal-Dual Algorithm

```
Input: Linear operator $\mathcal{K}$
       Preconditioning matrices $\mathcal{T}, \mathcal{S}$ according to (5.1)
       Proximity maps $\operatorname{prox}_{\mathcal{T},\mathcal{G}}, \operatorname{prox}_{\mathcal{S},\mathcal{F}^*}$
       Overrelaxation parameter $\gamma \in [0, 1)$

$x^0 = 0$, $y^0 = 0$                                                   ▷ Initialization
repeat
    $x^{l+1/2} = \operatorname{prox}_{\mathcal{T},\mathcal{G}}(x^l - \mathcal{T}\mathcal{K}^T y^l)$
    $y^{l+1/2} = \operatorname{prox}_{\mathcal{S},\mathcal{F}^*}(y^l + \mathcal{S}\mathcal{K}(2x^{l+1/2} - x^l))$
    $(x^{l+1}, y^{l+1}) = (x^{l+1/2}, y^{l+1/2}) + \gamma(x^{l+1/2} - x^l, y^{1+1/2} - y^l)$
until Termination criterion satisfied
```

The proximity operators are defined by

$$\begin{aligned}\operatorname{prox}_{\mathcal{T},\mathcal{G}}(\widehat{x}) &= \operatorname*{argmin}_x \frac{1}{2}\Big\langle \mathcal{T}^{-1}(x - \widehat{x}), x - \widehat{x} \Big\rangle + \mathcal{G}(x),\\ \operatorname{prox}_{\mathcal{S},\mathcal{F}^*}(\widehat{y}) &= \operatorname*{argmin}_y \frac{1}{2}\Big\langle \mathcal{S}^{-1}(y - \widehat{y}), y - \widehat{y} \Big\rangle + \mathcal{F}^*(y).\end{aligned}$$

To apply the algorithm all that is left is to specify the operator $\mathcal{K}$ and the proximity maps $\operatorname{prox}_{\mathcal{T},\mathcal{G}}$ and $\operatorname{prox}_{\mathcal{S},\mathcal{F}^*}$. Using the saddle-point formulation from the previous section this task is straightforward. As first termination criterion we check if an error estimator dropped below a prescribed threshold, and additionally limit the maximum number of iterations. We use a standard root-mean-squared (RMS) error estimator, separately for the primal and the dual variable

$$\operatorname{err}_{\mathrm{p}}(l) = \frac{\|x^l - x^{l-1}\|_2}{\sqrt{n}}, \quad \operatorname{err}_{\mathrm{d}}(l) = \frac{\|y^l - y^{l-1}\|_2}{\sqrt{m}}$$

and the following termination criterion

$$\operatorname{crit}_1(l) = \Big(\operatorname{err}_{\mathrm{p}}(l) < \mathrm{tol}_1\Big) \wedge \Big(\operatorname{err}_{\mathrm{d}}(l) < \mathrm{tol}_2\Big) \wedge \Big(l \leq l_{\max}\Big) \tag{5.2}$$

where l_{max} is set to a very high number and only serves to limit the runtime.

As a second criterion we additionally measure the primal and dual feasibility root-mean-squared error. Essentially when transforming a constrained optimization problem into the standard form of the primal-dual algorithm there are two types of constraints. The first one are simple constraints for which projecting onto the associated set of feasible points is simple and explicit. These constraints will always be satisfied after each iteration. The second type of constraints is usually expressed by introducing Lagrange multipliers. For these constraints we can measure the (root-mean-squared) primal and dual feasibility error by

$$\operatorname{err}_{\mathrm{pf}}(l) = \sqrt{\frac{\|\mathcal{E}_p x^l\|_2^2}{n}}, \quad \operatorname{err}_{\mathrm{df}}(l) = \sqrt{\frac{\|\mathcal{E}_d y^l\|_2^2}{m}}$$

where the matrix $\mathcal{E}_p$ contains all primal, and $\mathcal{E}_d$ all dual constraints of the second type. As alternative criterion we then use

$$\mathrm{crit}_2(l) = \mathrm{crit}_1(l) \wedge \Big(\mathrm{err}_{\mathrm{pf}}(l) < \mathrm{tol}_3\Big) \wedge \Big(\mathrm{err}_{\mathrm{df}}(l) < \mathrm{tol}_4\Big) \tag{5.3}$$

which, aside from estimating the optimality of the solution based on the convergence speed, in a sense permits control over the quality of the solution.

5.1.1 TVX0-PD

Casting TVX$_0$ from (4.9) into the standard form of the primal-dual algorithm requires only a few steps. With

$$m = \bar{M}\bar{N}K + 2MN, \quad n = \bar{M}\bar{N}K + MN$$

we set $x = (a, u) \in \mathbb{R}^n$, $y = (\psi, \varphi) \in \mathbb{R}^m$ and obtain $\mathcal{G}$ and $\mathcal{F}^*$ as

$$\begin{aligned} \mathcal{G}(x) &= \alpha \sum_{ijk} a^k_{ij} \, |\nu_k| + \mathbb{1}^{\infty}_{[0,\infty)^{MNK}}(a) + G(u), \\ \mathcal{F}^*(y) &= \mathbb{1}^{\infty}_{[-\frac{1}{2},\frac{1}{2}]^{MNK}}(\psi). \end{aligned}$$

The linear operator $\mathcal{K} \in \mathbb{R}^{m \times n}$ is then defined as

$$\mathcal{K} = \begin{pmatrix} \beta B & 0 \\ -D & A \end{pmatrix}.$$

The associated proximal maps are

$$\begin{aligned} x = \mathrm{prox}_{\mathcal{T},\mathcal{G}}(\widehat{x}) \iff a^k_{ij} &= \max(0, \widehat{a}^k_{ij} - \alpha \mathcal{T}^a_{ijk} \, |\nu_k|) \\ u &= \mathrm{prox}_{\mathcal{T}^u, G} \\ y = \mathrm{prox}_{\mathcal{S},\mathcal{F}^*}(\widehat{y}) \iff \psi_{ijk} &= \max(-\tfrac{1}{2}, \min(\tfrac{1}{2}, \widehat{\psi}_{ijk})) \\ \varphi_{ijk} &= \widehat{\varphi}_{ijk}. \end{aligned}$$

where with $\mathcal{T}^u$ and $\mathcal{T}^a$ we mean the sub-matrix of $\mathcal{T}$ that corresponds to u or a respectively. The proximity operator $\mathrm{prox}_{\mathcal{T}^u,G}$ depends on the data term of the specific image problem.

The matrix associated with measuring the primal-feasibility error is defined as

$$\mathcal{E}_p = \begin{pmatrix} -D & A \end{pmatrix}$$

and no explicit dual constrains are required.

5.1.2 TVX1-PD

To rewrite TVX$_1$ in the standard form of the primal-dual algorithm we define in analogy to the previous section

$$m = 2\bar{M}\bar{N}K + 2MN, \quad n = 2\bar{M}\bar{N}K + MN$$

and set $x = (a, v, u) \in \mathbb{R}^n$, $y = (\psi, \zeta, \varphi) \in \mathbb{R}^m$. From (4.11) the objective functions follow as

$$\begin{aligned} \mathcal{G}(x) &= \alpha \sum_{ijk} a^k_{ij} \, |\nu_k| + \mathbb{1}^{\infty}_{[0,\infty)^{MNK}}(a) + G(u), \\ \mathcal{F}^*(y) &= \mathbb{1}^{\infty}_{\left\{\widehat{\psi} \,|\, \sum_k \widehat{\psi}_{ijk}=0\right\}}(\psi) + \mathbb{1}^{\infty}_{[-1,1]^{MNK}}(\zeta) \end{aligned}$$

and the linear operator $\mathcal{K} \in \mathbb{R}^{m\times n}$ as

$$\mathcal{K} = \begin{pmatrix} \beta B & C^T & 0 \\ 0 & -\operatorname{Id} & 0 \\ -D & 0 & A \end{pmatrix}$$

where $\operatorname{Id} \in \mathbb{R}^{\bar{M}\bar{N}K \times \bar{M}\bar{N}K}$ denotes the identity matrix. The associated proximal maps are

$$\begin{aligned} x = \operatorname{prox}_{\mathcal{T},\mathcal{G}}(\widehat{x}) \iff a^k_{ij} &= \max(0, \widehat{a}^k_{ij} - \alpha \mathcal{T}^a_{ijk} \, |\nu_k|), \\ v_{ijk} &= \widehat{v}_{ijk}, \\ u &= \operatorname{prox}_{\mathcal{T}^u,G}(u), \\ y = \operatorname{prox}_{\mathcal{S},\mathcal{F}^*}(\widehat{y}), \iff \psi_{ijk} &= \widehat{\psi}_{ijk} - \frac{1}{K} \sum_k \widehat{\psi}_{ijk} \\ \zeta_{ijk} &= \max(-1, \min(1, \widehat{\zeta}_{ijk})), \\ \varphi_{ijk} &= \widehat{\varphi}_{ijk}, \end{aligned}$$

where as before $\mathcal{T}^u$ and $\mathcal{T}^a$ denote the sub-matrix of the preconditioning matrix $\mathcal{T}$ that belongs to u or a respectively. And again $\operatorname{prox}_{\mathcal{T}^u,G}$ depends on the specific data term $G(u)$.

The matrices associated with measuring the primal and dual feasibility error from the Lagrange constraints are defined as

$$\mathcal{E}_p = \begin{pmatrix} -D & 0 & A \end{pmatrix}, \quad \mathcal{E}_d = \begin{pmatrix} C & -\operatorname{Id} & 0 \end{pmatrix}.$$

5.2 Linear programming

In the following we will utilize the minimization formulations of TVX$_0$ and TVX$_1$ to rewrite optimization problem (3.24) as a linear program (LP), i.e. a minimization problem of the standard form

$$\begin{aligned} \min_x \quad & \frac{1}{2} c^T x \\ \text{subject to} \quad & \mathcal{A}_{eq} x = b_{eq}, \\ & \mathcal{A}_{iq} x \leq b_{iq}, \end{aligned}$$

where $c \in \mathbb{R}^n$, $A_{eq} \in \mathbb{R}^{p \times n}$ together with $b_{eq} \in \mathbb{R}^p$ define the equality constraints and $A_{iq} \in \mathbb{R}^{q \times n}$ with $b_{iq} \in \mathbb{R}^q$ the (component-wise) inequality constraints.

Of course this is only possible for linear data terms, which is why we add the requirement that the discrete data term G can be equivalently written in terms of a linear function

$$w \mapsto (c^G)^T w$$

and constraints

$$\mathcal{A}^G_{eq} w = b^G_{eq}, \qquad \mathcal{A}^G_{iq} w \leq b^G_{iq},$$

where $w = (v, u)^T \in \mathbb{R}^{V+MN}$ is a vector containing a set of (optional) auxiliary variables v and the image u, and $c^G \in \mathbb{R}^{V+MN}$ the linear coefficients for the objective function.

5.2.1 Generalization for quadratic data terms

Many of the example regularization problems from Section 3.7 provide non-linear data terms and hence the linear programming formulation is not sufficient. While many generalizations of an LP are available it is sufficient to consider convex quadratic programming (QP) to cover all example problems. The standard form of a QP is the minimization problem

$$\begin{aligned} \min_x \quad & \frac{1}{2} x^T Q x + c \cdot x \\ \text{subject to} \quad & \mathcal{A}_{eq} x = b_{eq} \\ & \mathcal{A}_{iq} x \leq b_{iq} \end{aligned}$$

where $Q \in \mathbb{R}^{n \times n}$ is positive semidefinite and symmetric, and as with the LP formulation $c \in \mathbb{R}^n$, $A_{eq} \in \mathbb{R}^{p \times n}$ together with $b_{eq} \in \mathbb{R}^p$ define the equality constraints and $A_{iq} \in \mathbb{R}^{q \times n}$ with $b_{iq} \in \mathbb{R}^q$ the (component-wise) inequality constraints. Strictly speaking the requirement of Q being positive semidefinite is unnecessary, but it ensures convexity of the problem.

Using this extension any image-problem of the form (3.24) can be solved if the discrete data term G can be equivalently written in terms of a quadratic function

$$w \mapsto \frac{1}{2} w^T Q^G w + c^G \cdot w$$

and constraints

$$\begin{aligned} \mathcal{A}_{eq}^G w &= b_{eq}^G, \\ \mathcal{A}_{iq}^G w &\leq b_{iq}^G \end{aligned}$$

where with $n = V + MN$

- $w = (v, u)^T \in \mathbb{R}^n$ is a vector containing a set of (optional) auxiliary variables v and the image u,
- $Q^G \in \mathbb{R}^{n \times n}$ is a positive semidefinite and symmetric matrix,
- $c^G \in \mathbb{R}^n$ the linear coefficients for the objective functions.

To generalize optimization problems TVX0-LP and TVX1-LP from the previous section to accept a quadratic data-term all we have to do is to define the matrix $Q \in \mathbb{R}^{n \times n}$ as

$$Q = \begin{pmatrix} 0 & 0 \\ 0 & Q^G \end{pmatrix}.$$

These modified functionals are then called **TVX0-QP** and **TVX1-QP** respectively.

5.2.2 Transformation of norm minimization problems

During the numerical implementation it became apparent that the choosen method for transforming norm minimization problems into linear or quadratic programs had a strong impact on computation time and memory consumption. While on their own they could be solved in almost no time, there was a notable difference between the formulations when used in combination with other constraints and a more complex objective function. This section will present different choices for the transformation, as no single most efficient variant could be determined through experiments with different solvers.

For $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $p \in \{1, 2\}$ consider the minimization problem

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_p^p.$$

If $p = 1$ then the problem is equivalent to the LP

$$\begin{aligned} \min_{x \in \mathbb{R}^n,\, t \in \mathbb{R}^m} \quad & \sum_{i=1}^m t_i \\ \text{subject to} \quad & +(Ax - b) \leq t, \\ & -(Ax - b) \leq t. \end{aligned}$$

Alternatively, by introducing slack variables this can be transformed to the equivalent problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n,\, t,s\in\mathbb{R}^m} \quad & \sum_{i=1}^m t_i \\ \text{subject to} \quad & +(Ax-b)-t+s=0, \\ & -(Ax-b)\leq t, \\ & s\geq 0, \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \min_{x\in\mathbb{R}^n, s\in\mathbb{R}^m} \quad & \sum_{i=1}^m (Ax-b)_i + s_i \\ \text{subject to} \quad & -2(Ax-b)-s\leq 0, \\ & s\geq 0, \end{aligned}$$

which has half the number of general inequality constraints and a set of simple non-negativity constraints. The constant $-\sum_{i=1}^m b_i$ in the objective can further be omitted.

As expected, if A is sparse with only very few non-zero elements, such as an identity matrix, there was no significant difference. However with growing complexity the memory requirement for the first formulation was notably higher. And since memory turned out to be one of the major bottlenecks during the numerical computations the latter approach was implemented.

For $p=2$ we can utilize the following transformations.

$$\begin{aligned} \min_{x\in\mathbb{R}^n,\, s\in\mathbb{R}^m} \quad & s^T s \\ \text{subject to} \quad & Ax-b=s \end{aligned}$$

hence the matrix $Q\in\mathbb{R}^{n+m\times n+m}$ of the quadratic function is a simple sparse block matrix

$$Q=\begin{pmatrix} 0 & 0 \\ 0 & \mathrm{Id}_n \end{pmatrix}$$

which is of course positive semidefinite. Another way to restate this problem is

$$\min_{x\in\mathbb{R}^n} x^T A^T Ax - 2x^T A^T b + b^T b$$

where $b^T b$ is constant and thus can be omitted. Therefore we can rewrite the problem as

$$\min_{x\in R^n} x^T Qx + 2c^T x$$

with $c=-A^T b$ and $Q=A^T A$, which is again positive and semidefinite. Up until rescaling this coincides with the standard form of a quadratic program. Interestingly in many cases the first formulation performed notably better in numerical experiments.

If the available solver supports (second-order) cone constraints then exploring the following might also be of interest. In particular this is the form that was implemented for the examples, as the more general second-order cone programming outperformed the quadratic programming by length. This is only a technical restriction on the solver that might be resolved at any time. Therefore we will not go into too much detail at this point and refer to literature on second-order cone programming (SOCP) such as [1] or [6]. The standard form of the SOCP in consideration is

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{subject to} \quad & \mathcal{A}_{eq} x = b_{eq}, \\ & \mathcal{A}_{iq} x \leq b_{iq}, \\ & (x_{i_1}, \dots, x_{i_{n_i}}) \in C_i \text{ for } i = 1, \dots, c \end{aligned} \tag{5.4}$$

where $c \in \mathbb{N}_0$ is the number of cones, $(x_{i_1}, ..., x_{i_{n_i}})$ a subsequence of x belonging to cone C_i of length $n_i \in \mathbb{N}$. Although this is not the most concise form, as inequality constraints can easily be written in terms of conic constraints, it ensures compatibility to the LP form whenever $c = 0$.

Now, first notice that the epigraph of the 2-norm can be written as

$$\operatorname{epi} \|\cdot\|_2 = \left\{ (x, \lambda) \in \mathbb{R}^{n+1} \;\middle|\; \lambda \geq \|x\|_2 \right\} = \left\{ (x, \lambda) \in \mathbb{R}^{n+1} \;\middle|\; \lambda^2 \geq \|x\|_2^2, \lambda \geq 0 \right\} = \mathcal{C}^{n+1}$$

which is of course convex and is called the quadratic or second-order cone for the Euclidean norm. Using a linear transformation the cone $\mathcal{C}^{n+2}$ can be transformed into

$$\mathcal{C}_r^{n+2} = \left\{ (x, \lambda_1, \lambda_2) \in \mathbb{R}^{n+2} \;\middle|\; 2\lambda_1 \lambda_2 \geq \|x\|^2, \lambda_1, \lambda_2 \geq 0 \right\}$$

see for example [1, Section 2.3]. For $x \in \mathbb{R}^n$ and $t \geq 0$ the squared 2-norm may be described by

$$\|x\|_2^2 \leq t \Leftrightarrow (x, \frac{1}{2}, t) \in \mathcal{C}_r^{n+2}$$

and the squared-2 norm minimization problem as SOCP is

$$\begin{aligned} \min_{(x,f,t) \in \mathbb{R}^{n+m+1}} \quad & t \\ \text{subject to} \quad & Ax - b = f, \\ & (f, 1/2, t) \in \mathcal{C}_r^{m+2}. \end{aligned}$$

Remark 5.1. *As already pointed out before, choosing the equivalent formulation that performs best highly depends on the solver being used. This is especially the case for the reformulation into the more general SOCP, which mathematically speaking, is unnecessary. And while the restriction is purely technical, selecting the appropriate formulation determined whether a problem was computable within a few minutes or a few days at best.*

In light of the last section we will also provide implementations for second-order cone programming of the standard form (5.4) and name the optimization problems **TVX0-CP** and **TVX1-CP** whenever they are subject to cone constraints.

5.2.3 TVX0-LP

Together with the last section, recasting the minimization formulation (4.10) of TVX$_0$ into the standard form of a linear program is straightforward. Let $n = 2\bar{M}\bar{N}K + V + MN$ and set

$x = (a, s, v, u) \in \mathbb{R}^n$. Then the linear objective is determined by $c \in \mathbb{R}^n$

$$c = \Big(\alpha(|\nu_k|)_{ijk} + \frac{\beta}{2}\sum_{i=1}^{\bar{M}\bar{N}K}(B_{i,1}, \dots, B_{i,\bar{M}\bar{N}K}),\ \frac{\beta}{2}\mathbf{1},\ c^G\Big)$$

where $\mathbf{1} = (1, \dots, 1) \in \mathbb{R}^{\bar{M}\bar{N}K}$ is a vector of ones. The equality constraints are defined by

$$\mathcal{A}_{eq} = \begin{pmatrix} -D & 0 & 0 & A \\ 0 & 0 & \multicolumn{2}{|c}{\mathcal{A}^G_{eq}} \end{pmatrix}, \quad b_{eq} = \begin{pmatrix} 0 \\ b^G_{eq} \end{pmatrix}$$

and the inequality constraints by

$$\mathcal{A}_{iq} = \begin{pmatrix} -\operatorname{Id} & 0 & 0 \\ 0 & -\operatorname{Id} & 0 \\ -2B & -\operatorname{Id} & 0 \\ 0 & 0 & \mathcal{A}^G_{iq} \end{pmatrix}, \quad b_{iq} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ b^G_{iq} \end{pmatrix}$$

where $\operatorname{Id} \in \mathbb{R}^{\bar{M}\bar{N}K \times \bar{M}\bar{N}K}$ denotes the identity matrix.

5.2.4 TVX1-LP

Using the minimization formulation (4.12) of TVX_1 the transformation to a linear program is similar to the previous section. Let $n = 3\bar{M}\bar{N}K + V + MN$ and define $x = (a, s, p, v, u) \in \mathbb{R}^n$. The objective is defined by $c \in \mathbb{R}^n$

$$c = \Big(\alpha(|\nu_k|)_{ijk},\ \beta\mathbf{1},\ c^G\Big)$$

where $\mathbf{1} = (1, \dots, 1) \in \mathbb{R}^{2\bar{M}\bar{N}K}$ again denotes a vector of ones. Moreover, the equality constraints are identified by

$$\mathcal{A}_{eq} = \begin{pmatrix} -D & 0 & 0 & 0 & A \\ B & 0 & C^T & 0 & 0 \\ 0 & 0 & 0 & \multicolumn{2}{|c}{\mathcal{A}^G_{eq}} \end{pmatrix}, \quad b_{eq} = \begin{pmatrix} 0 \\ 0 \\ b^G_{eq} \end{pmatrix}$$

and the inequality constraints by

$$\mathcal{A}_{iq} = \begin{pmatrix} -\operatorname{Id} & 0 & 0 & 0 \\ 0 & -\operatorname{Id} & -2\operatorname{Id} & 0 \\ 0 & 0 & -\operatorname{Id} & 0 \\ 0 & 0 & 0 & \mathcal{A}^G_{iq} \end{pmatrix}, \quad b_{iq} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ b^G_{iq} \end{pmatrix}$$

where as before $\operatorname{Id} \in \mathbb{R}^{\bar{M}\bar{N}K \times \bar{M}\bar{N}K}$ denotes the identity matrix.

6 Applications and Evaluation

This chapter will discuss numerical results of the new discretization for TVX_0 and TVX_1 using the examples from Section 3.7. Aside from presenting differences between the two regularization functionals, a comparison between the new discretization and the original one proposed in [7] will be made. The last section then focuses on differences between the implementation using linear or quadratic programming and the primal-dual algorithm, by comparing performances and evaluating practical considerations.

6.1 Numerical results

6.1.1 Noise removal

The first example is the task of noise removal as proposed in Section 3.7.1. For Gaussian noise we consider an ℓ^2 data fidelity term

$$G(u) = \frac{\lambda}{2} \|u - f\|_2^2$$

where, as before, $f \in \mathbb{R}^{MN}$ denotes the noisy image and $\lambda > 0$ is a tuning parameter. To apply the primal-dual algorithm we need to specify the problem-specific proximal-map, which is given by

$$u = \mathrm{prox}_{\mathcal{T}^u, G}(\widehat{u}) \Leftrightarrow u_{ij} = \frac{\widehat{u}_{ij} + \lambda \mathcal{T}^u_{ij} f_{ij}}{1 + \lambda \mathcal{T}^u_{ij}}.$$

For QP we need to specify the quadratic function. We set $w = u \in \mathbb{R}^{MN}$ and following Section 5.2.2 the data term is fully determined by

$$c^G = -\lambda f, \quad Q^G = \lambda \,\mathrm{Id}$$

To apply SOCP we set $w = (s, t, v, u) \in \mathbb{R}^{2+2MN}$ and then the objective is determined by

$$c^G = (0, 1, 0, \dots, 0)$$

subject to constraints

$$\mathcal{A}^G_{eq} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -\mathrm{Id} & \mathrm{Id} \end{pmatrix}, \quad b^G_{eq} = \begin{pmatrix} \frac{1}{2} \\ f \end{pmatrix}, \quad (v, s, t) \in \mathcal{C}_r^{MN+2}.$$

For impulse noise an ℓ_1 data fidelity term has proven to perform better. Hence we consider

$$G(u) = \lambda \left\| u - f \right\|_1 .$$

The proximity operator is given by the shrinkage formula

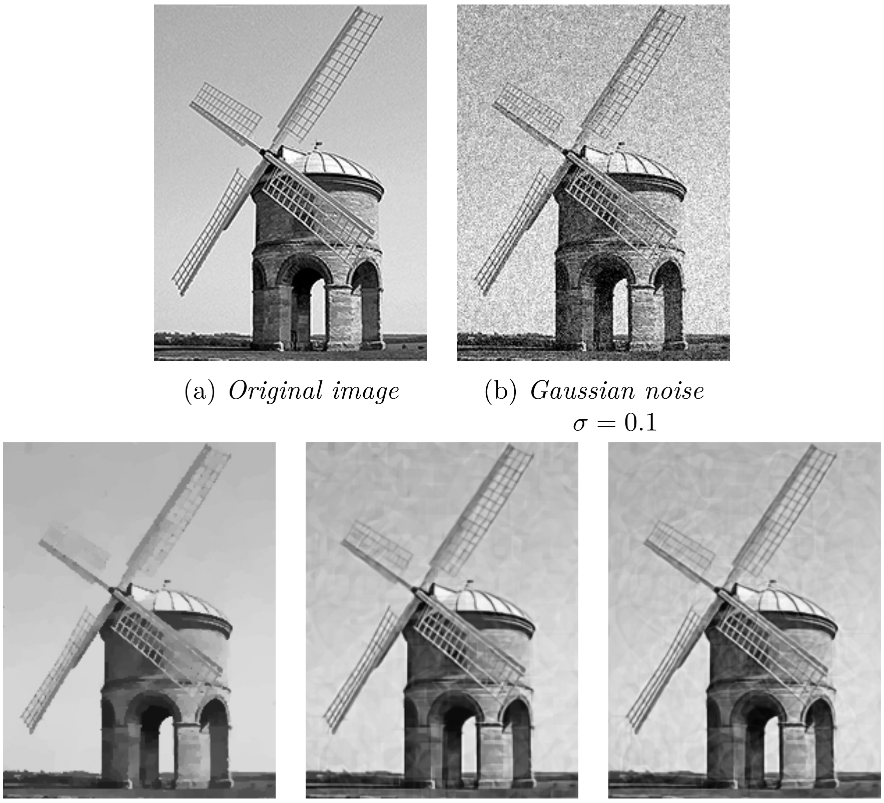
$$u = \operatorname{prox}_{\mathcal{T}^u, G}(\widehat{u}) \iff u_{i,j} = \begin{cases} \widehat{u}_{i,j} - \lambda \mathcal{T}^u_{i,j} & \text{if} \quad \widehat{u}_{i,j} - f_{i,j} > \quad \lambda \mathcal{T}^u_{i,j}, \\ \widehat{u}_{i,j} + \lambda \mathcal{T}^u_{i,j} & \text{if} \quad \widehat{u}_{i,j} - f_{i,j} < -\lambda \mathcal{T}^u_{i,j}, \\ f_{i,j} & \text{if} \quad |\widehat{u}_{i,j} - f_{i,j}| \leq \quad \lambda \mathcal{T}^u_{i,j}. \end{cases}$$

This problem can of course be solved using a linear program. With $w = (v, u) \in \mathbb{R}^{2MN}$ the constraints are defined by

$$\mathcal{A}^G_{iq} = \begin{pmatrix} -\operatorname{Id} & -2\operatorname{Id} \\ -\operatorname{Id} & 0 \end{pmatrix}, \quad b^G_{iq} \begin{pmatrix} -2f \\ 0 \end{pmatrix}$$

and the linear objective is determined by

$$c^G = \lambda(1, \ldots, 1).$$


(a) *Original image*

(b) *Gaussian noise* $\sigma = 0.1$

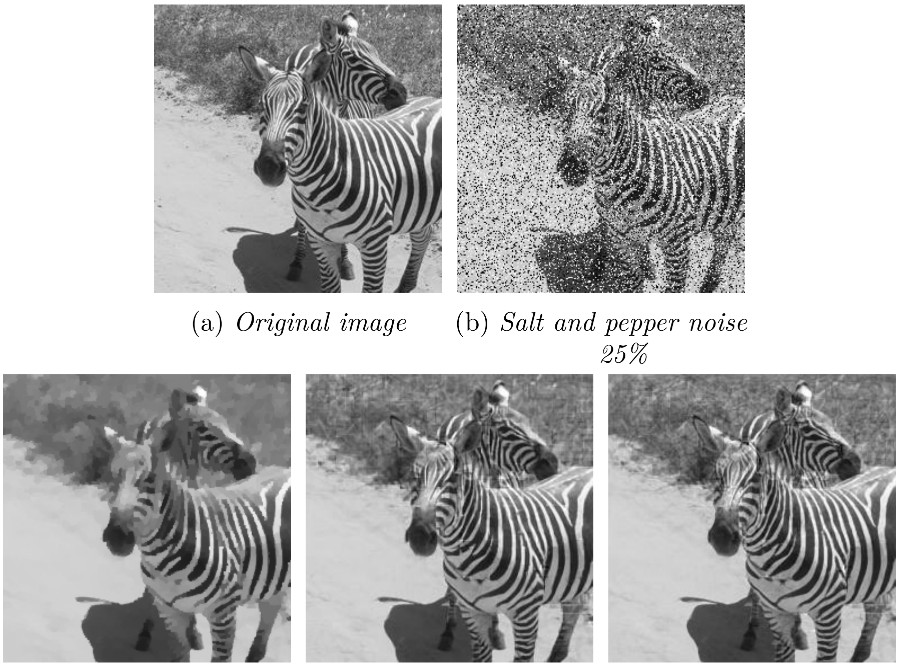
(c) TV-LP PSNR = 21.375

(d) TVX0-CP PSNR = 22.55

(e) TVX1-CP PSNR = 23.09

Figure 6.1: *Results of noise removal using an ℓ^2 data fidelity measure. Parameters for* TVX-CP *are $K = 16$, $\alpha = 0.1$, $\beta = 1.0, \lambda = 5$. For the* TV *regularization the weighting parameter was set to $\lambda = 1$.*

Figure 6.1 exhibits the results of Gaussian noise removal using the different functionals. While TV-regularization works well for restoring homogeneous regions such as the sky, the line structures of the sails are mostly discontinuous or blurred. As expected TVX_0 and TVX_1 are much better in preserving these line structures. However both functionals tend to generate directional artifacts in homogeneous regions.



(a) *Original image* (b) *Salt and pepper noise 25%*

(c) TV-LP
PSNR = 21.02

(d) TVX0-LP
PSNR = 24.25

(e) TVX1-LP
PSNR = 24.72

Figure 6.2: *Results of noise removal using an ℓ^1 data fidelity measure. Parameters for* TVX-LP *are $K = 16$, $\alpha = 0.1$, $\beta = 1.0, \lambda = 0.75$. For the* TV *regularization the weighting parameter was set to $\lambda = 0.125$.*

Figure 6.2 displays the results of noise removal for salt and pepper noise. The images computed using curvature regularization show a very successful reconstruction of the zebra's contours, while using TV-regularization leads to shrinkage at the ears and legs. Also the striping patterns are much sharper for TVX_0 and TVX_1. At a close look for TVX_0 one can observe a few rectangular patches that may have been introduced by a disadvantageous local constellation of noise and zebra pattern forming line-like structures.

6.1.2 Binary segmentation

Next we consider the task of binary segmentation as in Section 3.7.2. The corresponding data term is given by

$$G(u) = \lambda \sum_{i,j} f_{ij} u_{ij} + \mathbb{1}^{\infty}_{[0,1]^{MN}}(u)$$

with tuning parameter $\lambda > 0$ and an external segmentation field f. The problem-specific proximity operator required for the primal-dual algorithm can be identified as

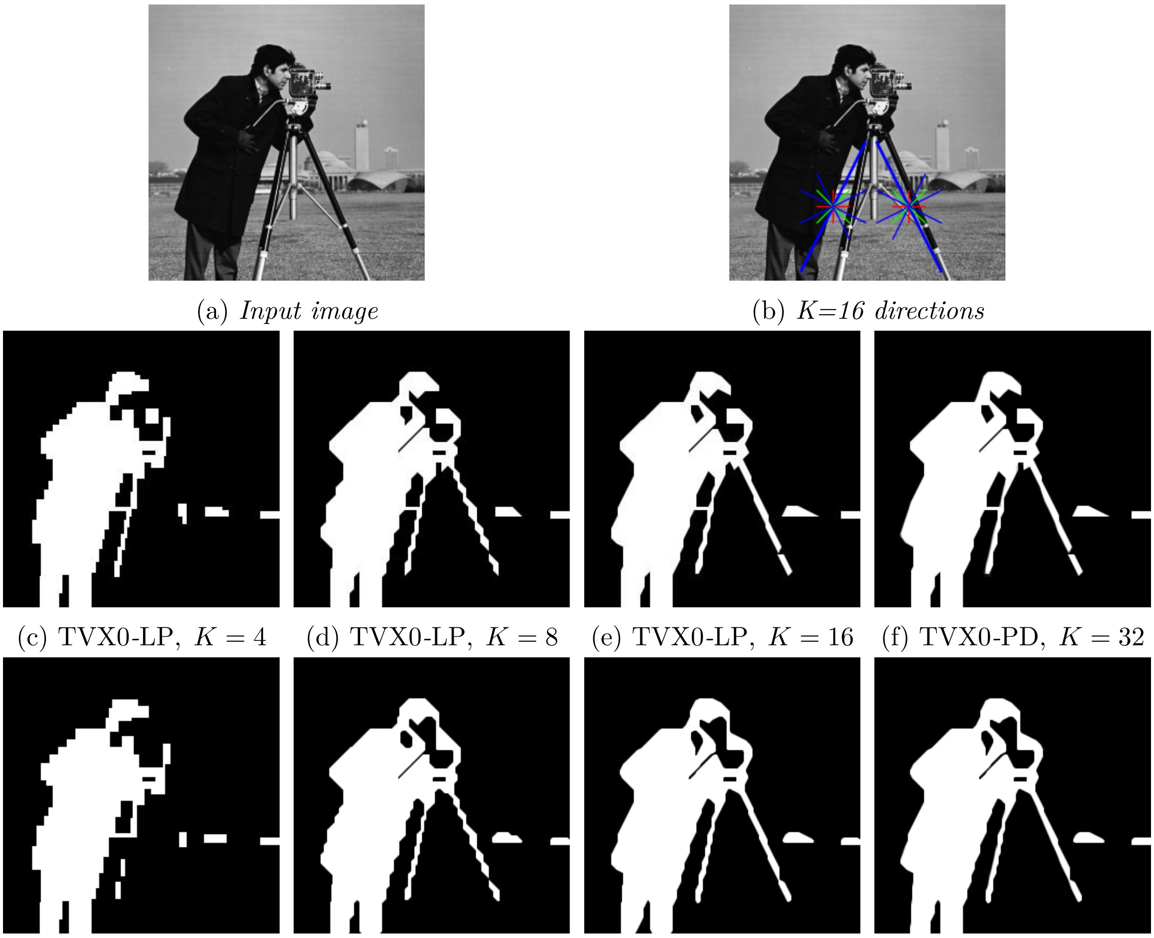
$$u = \mathrm{prox}_{\mathcal{T}^u,G}(\widehat{u}) \iff u_{i,j} = \max(0, \min(1, \widehat{u}_{i,j} - \lambda \mathcal{T}^u_{i,j} f_{i,j})).$$

A linear program is sufficient to solve this problem. We set $w = u \in \mathbb{R}^{MN}$, i.e. no additional auxiliary variables are required. The objective function is determined by $c^G = f$, and the optimization is subject to inequality constraints

$$\mathcal{A}^G_{iq} = \begin{pmatrix} -\mathrm{Id} \\ \mathrm{Id} \end{pmatrix}, \quad b^G_{iq} = \begin{pmatrix} 0 \\ \mathbf{1} \end{pmatrix}.$$

In the first example the segmentation field was computed as follows. For a discrete input image $\mathcal{I} \in \mathbb{R}^{MN}$ and μ_f, μ_b denoting the mean foreground and mean background value respectively, f is computed as

$$f_{ij} = (\mathcal{I}_{ij} - \mu_f)^2 - (\mathcal{I}_{ij} - \mu_b)^2.$$



(a) *Input image*

(b) *K=16 directions*

(c) TVX0-LP, $K = 4$

(d) TVX0-LP, $K = 8$

(e) TVX0-LP, $K = 16$

(f) TVX0-PD, $K = 32$

(g) TVX1-LP, $K = 4$

(h) TVX1-LP, $K = 8$

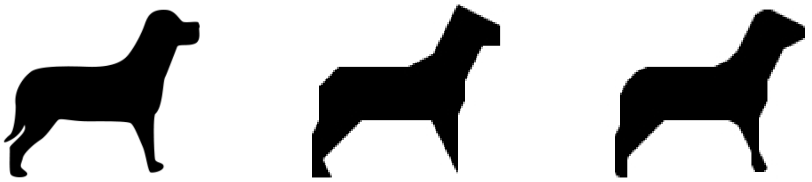
(i) TVX1-LP, $K = 16$

(j) TVX1-LP, $K = 32$

Figure 6.3: *Comparison of results for* TVX_0 *and* TVX_1 *for different numbers of discrete orientations. In all images parameters were chosen as* $\alpha = 0.1$, $\beta = 1$, $\lambda = 1$ *and the mean foreground and background values were set to* $\mu_f = 0$, $\mu_b = 0.5$.

Figure 6.3 exhibits the results of the TVX regularization for a varying number of discrete orientations K. At first glance both functionals TVX_0 and TVX_1 produce similar results. For $K = 4$ the strong directional preference is clearly visible. By increasing the number of directions to $K = 8$ this effect is slightly attenuated although still very prominent along the tripod.

Because the selected directions by design have to be aligned to the discrete grid, the regularization introduces a minor directional bias as the orientations are not uniformly distributed for $K = 16$. Image (b) demonstrates that neither the camera man's jacket nor the camera stand are perfectly aligned to the any of the 16 directions, which results in a 'stair-casing' approximation. To compute the images for $K = 32$ directions we had to utilize the primal-dual approach due to memory restrictions. Although the number of discrete orientations is very high the results still resemble directional artifacts. These findings are different than the results using the original discretization, which at a higher number of discrete orientations smoothly approximated all boundaries without directional artifacts. While, as expected, TVX_0 in general prefers polygonal shapes with sharp edges, TVX_1 yields smoother edges, which can for example be observed at the neck and head area of the camera man.



(a) *Input image* (b) TVX0-LP (c) TVX1-LP

Figure 6.4: *Comparison of* TVX_0 *and* TVX_1 *for strong curvature regularization. Parameters:* $\alpha = 0.0, \beta = 2, \lambda = 0.25$ *and* $K = 16$.

Figure 6.4 further demonstrates that the two regularization functionals work as expected. While TVX_0 approximates the shape through a simplified polygon, thus minimizes the number of vertices, TVX_1 tends to preserve round areas, and eliminates concave parts that would increase the total curvature.

In a later section we will further explore the effects of varying the weighting between regularization and the data term. These results will then be used to compare the initial discretization from [7] to the one proposed in this thesis.

6.1.3 Inpainting

As a third example we consider inpainting as in Section 3.7.3. Consider $\mathcal{I} \in \mathbb{R}^{MN}$ to be an incomplete image, where the pixel values are only known on a non-empty indexing set $\Omega' \subset \{0, \dots, M-1\} \times \{0, \dots, N-1\}$. Then the data term is given by

$$G(u) = \sum_{(i,j)\in\Omega'} \begin{cases} 0 & \text{if } u_{ij} = \mathcal{I}_{ij}, \\ \infty & \text{otherwise.} \end{cases}$$

To apply the primal-dual algorithm we again need to specify the problem-specific proximity operator, which is determined by

$$u = \operatorname{prox}_{\mathcal{T}^u,G}(\widehat{u}) \iff u_{i,j} = \begin{cases} \mathcal{I}_{i,j} & \text{if } (i,j) \in \Omega', \\ \widehat{u}_{i,j} & \text{otherwise.} \end{cases}$$

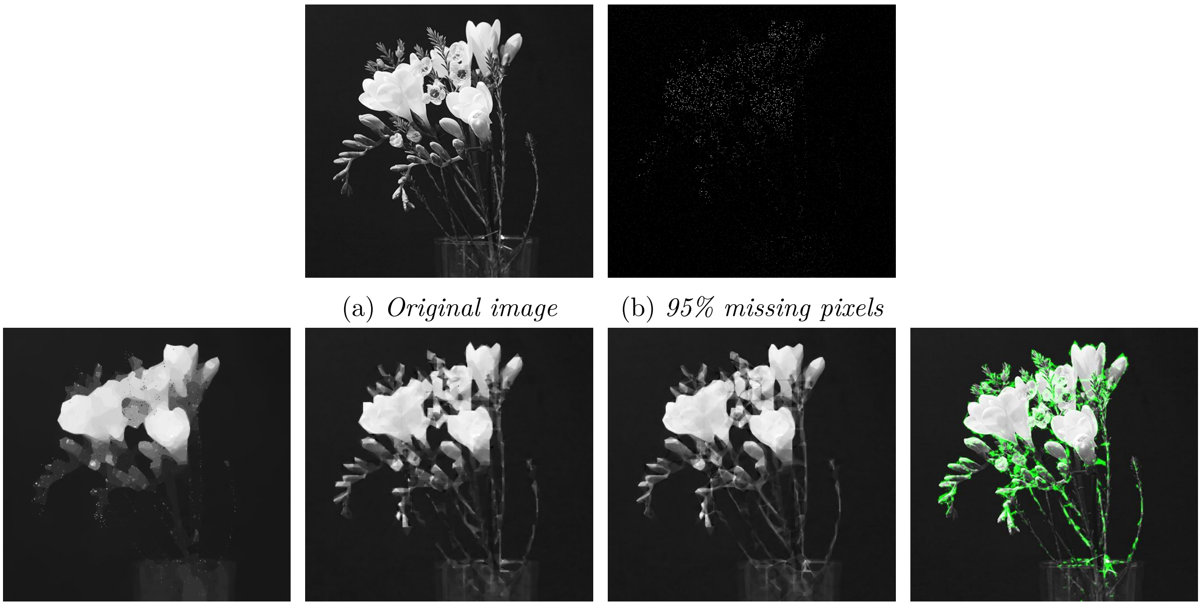
This problem can easily be transformed into a linear program. We define $P \in \mathbb{R}^{MN}$ as

$$P_{ij} = \begin{cases} 1 & (i,j) \in \Omega', \\ 0 & \text{otherwise,} \end{cases}$$

and $w = u \in \mathbb{R}^{MN}$. The equality constraints are then given by

$$\mathcal{A}^G_{eq} = \operatorname{diag}(P), \quad b^G_{eq} = \operatorname{diag}(P)\mathcal{I}$$

and the objective function $c^G = 0$.



(a) *Original image* (b) *95% missing pixels*

(c) TV-LP PSNR = 19.9655

(d) TVX0-PD PSNR = 22.1855

(e) TVX1-PD PSNR = 21.9835

(f) *Differences* TVX_0 TVX_1 *larger than 5%.*

Figure 6.5: *Results of inpainting using different regularization functionals. Parameters for* TVX-PD *regularization are* $K = 16$, $\alpha = 0.1$, $\beta = 1.0$.

Following the inpainting example from [7] we removed 95% of all pixels and then applied the different regularization functionals to restore the missing information. The first image displays the result of a total variation regularization, which was computed by setting the regularization parameter $\beta = 0$ and then applying linear programming.

Computing TVX0-LP and TVX1-LP was unsuccessful due to memory limitations. Thus both result images were computed using the primal-dual algorithm.

The curvature regularization functionals are much better at completing the object boundaries and reconstruction of thin, elongated structures. The length-based regularization on the other hand eliminated almost all details in favor of uniformly colored patches. TVX_1 yields notably smoother boundaries than TVX_0 and Figure (f) further illustrates that the major differences between the two functionals are located along the objects boundaries.

6.1.4 Deconvolution

As a last example of general image problems we consider deconvolution as in Section 3.7.4. Let $\mathcal{I} \in \mathbb{R}^{MN}$ be a blurry and noisy image. We assume that the point spread function $\mathrm{PSF} \in \mathbb{R}^{MN\times MN}$ is known. If $\mathcal{I}$ is subject to Gaussian noise we consider the data term

$$G(u) = \frac{\lambda}{2} \left\|\mathrm{PSF}\, u - \mathcal{I}\right\|_2^2$$

where $\lambda > 0$ is a tuning parameter. We only implemented this system using quadratic programming, but the proximal maps can be found in [8, Section 4.2.3]. The transformation into a QP is similar to the classical Gaussian noise removal case. Hence with $w = (v, u) \in \mathbb{R}^{2MN}$ the equality constraints and quadratic function are defined by

$$A_{eq}^G = \begin{pmatrix} \mathrm{Id} & \mathrm{PSF} \end{pmatrix}, \quad b_{eq}^G = \mathcal{I}, \qquad c^G = 0, \quad Q^G = \lambda \begin{pmatrix} \mathrm{Id} & 0 \\ 0 & 0 \end{pmatrix}.$$

Similarly for impulse noise we consider $G(u) = \lambda \left\|\mathrm{PSF}\, u - \mathcal{I}\right\|$. Again with $w = (m_1, m_2, u) \in \mathbb{R}^{3MN}$ we have constraints

$$\mathcal{A}_{eq}^G = \begin{pmatrix} 0 & \mathrm{Id} & \mathrm{PSF} \end{pmatrix}, \quad b_{eq}^G = \mathcal{I}, \qquad \mathcal{A}_{iq}^G = \begin{pmatrix} -\mathrm{Id} & \mathrm{Id} & 0 \\ -\mathrm{Id} & -\mathrm{Id} & 0 \end{pmatrix}, \quad b_{iq}^G = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

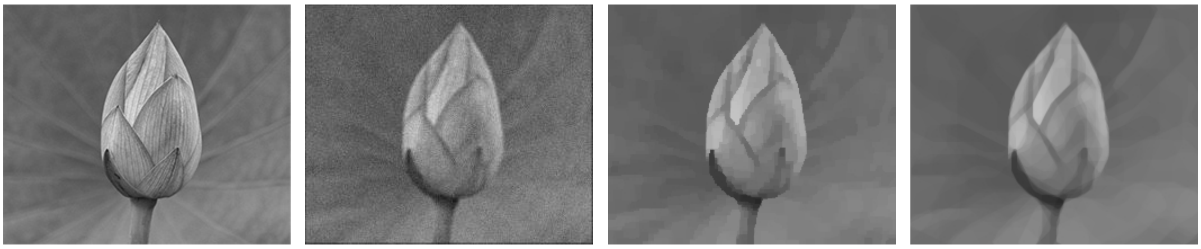

(a) *Original image* (b) *Blurry and noise input* (c) TV-LP PSNR = 31.79 (d) TVX1-LP PSNR = 31.37

Figure 6.6: *Results of deconvolution using different regularization functionals. The input image was degraded using a Gaussian blur kernel of size* 4×4 *with standard deviation* $\sigma = 3$ *and Gaussian noise with* $\sigma^2 = 0.01$*. Parameters for* TVX1-LP *regularization are* $K = 16$*,* $\alpha = 0.1$*,* $\beta = 1.0$*,* $\lambda = 5$*.*

Figure 6.6 shows the results of the proposed deconvolution with length-based and curvature regularization. Both functionals are unable to condense the contours back to the original thin form. However, while the length-based regularization produces pixelated lines and non-smooth gradients with a similar effect as quantizing the gray levels, the result of curvature regularization in general is much smoother. The downside of the image's smoothness is a loss of perceptive sharpness. The result of TVX_0 regularization was omitted as it showed no visible difference to the TVX_1 regularization.

6.1.5 Thin structures preserving noise removal

The final application in consideration is the specialized noise removal for thin, line-structure images with balanced flow as described in Section 3.7.5. We will only provide an implementation using linear or quadratic programming.

Modified regularization functionals

To apply TVX_0 or TVX_1 regularization to the specialized noise removal problems we have to slightly adjust the discrete formulations of TVX by replacing the compatibility condition with the flow-balancing condition. With the remarks from Section 5.2.2 about norm minimization problems the transformation into the appropriate standard form of the solver is straightforward.

For a quadratic data term $G : \mathbb{R}^{MN} \to [0, \infty)$ according to Theorem 3.26 the **discrete** $\mathrm{TVX}_\mathbf{0}$**-regularization** of G is defined as

$$\begin{aligned} \min_{a,s} \qquad & \alpha \sum_{ijk} a_{ij}^k \, |\nu_k| + \frac{\beta}{2} \, \|Ba\|_1 + G(Pa) \\ \text{subject to} \quad & a_{ij}^k \geq 0, \; Fa = 0. \end{aligned}$$

and the **discrete** $\mathrm{TVX}_\mathbf{1}$**-regularization** of G as

$$\begin{aligned} \min_{a,s} \qquad & \alpha \sum_{ijk} a_{ij}^k \, |\nu_k| + \beta \, \|p\|_1 + G(Pa) \\ \text{subject to} \quad & a_{ij}^k \geq 0, \; Fa = 0, \; C^T p + Ba = 0. \end{aligned}$$

The next step is to formulate the discrete data terms for the Wasserstein and the H^{-1} distances.

Discrete Wasserstein-1 metric

In this section measures $v \in \mathcal{M}(\Omega)$ are approximated by a sum of delta peaks, i.e.

$$v \approx \sum_{ij} v_{ij} \delta_{x_{ij}} \tag{6.1}$$

where as usual $\delta_{x_{ij}}$ denotes the delta distribution at $x_{ij} = (ih, jh) \in \Omega_h$ and v_{ij} are linear coefficients. For two probability measures $\mu^1, \mu^2 \in \mathcal{P}(\Omega)$ the Wasserstein-1 metric, according to 2.9, can be computed as

$$\mathrm{dist}_{W^1}(\mu^1, \mu^2) = \sup \left\{ \int_\Omega \varphi \, \mathrm{d}(\mu^1 - \mu)^2 \;\middle|\; \varphi : \Omega \to \mathbb{R}, \text{ 1-Lipschitz} \right\}.$$

Hence for μ^1, μ^2 discretized according to (6.1) and with Proposition 2.26

$$\mathrm{dist}_{W^1}(\mu^1, \mu^2) \approx \sup_{\substack{\varphi \in \mathbb{R}^n \\ \|A\varphi\|_\infty \leq 1}} \sum_{ij} \varphi_{ij} (\mu_{ij}^1 - \mu_{ij}^2) = \min_{\substack{\varphi \in \mathbb{R}^{2n} \\ (A^T \varphi)_{ij} + (\mu_{ij}^1 - \mu_{ij}^2) = 0}} \|\varphi\|_1 .$$

Observe that the definition of the W_1-metric expects two probability measures. We want to use it to compute distances to given real-valued, positive finite measures that represent the density of the special line-structure. These measures are not necessarily probability measures, but the only property in question is the normalization. Instead of rescaling the measures to be normalized, we choose to ensure that we only compute the W_1 distance between (positive) measures of the same total mass, which results in an additional constraint. In summary the data term for W_1-noise removal reads as follows.

Given an input density $\mathcal{I} \in \mathbb{R}^{MN}$, defined by its linear coefficients according to (6.1), the associated discrete data term $G_{W_1} : \mathbb{R}^{MN} \to [0, \infty)$ is

$$\begin{aligned} G_{W_1}(Pa) = \min_{\varphi \in \mathbb{R}^{2MN}} \quad & \|\varphi\|_1 + \mathbb{1}^{\infty}_{\{\|Pa\|_1 = \|\mathcal{I}\|_1\}} \\ \text{subject to} \quad & A^T\varphi + (Pa - \mathcal{I}) = 0. \end{aligned}$$

Discrete H^{-1} metric

According to Definition 2.14 the squared H^{-1} distance between two measures $\mu^1, \mu^2 \in \mathcal{M}(\Omega) \cap H^{-1}(\Omega)$ can be computed by

$$\operatorname{dist}^2_{H^{-1}}(\mu^1, \mu^2) = \left\| \nabla(-\Delta)^{-1}(\mu^1 - \mu^2) \right\|^2_{L^2}.$$

Because the operator $\nabla(-\Delta)^{-1}$ is difficult to characterize and acts non-locally we compute the norm in two steps. First we solve

$$\begin{cases} -\Delta f = \mu^1 - \mu^2 & \text{in } \Omega, \\ \quad f = 0 & \text{on } \partial\Omega \end{cases} \tag{i}$$

and then compute $\|\nabla(-\Delta)^{-1}(\mu^1 - \mu^2)\|^2_{L^2} = \|\nabla f\|^2_{L^2}$. Discrete approximations of μ^1 and μ^2 have to be contained in $\mathcal{M}(\Omega) \cap H^{-1}(\Omega)$ hence we select the following discretization

$$\mu^m \approx \mu^m_h = \sum_{ij} \mu^m_{ij} \frac{1}{\mathcal{L}^2(U_{ij})} \mathcal{L}^2 \llcorner U_{ij} = \frac{1}{h^2} \sum_{ij} \mu^m_{ij} \mathcal{L}^2 \llcorner U_{ij} \tag{6.2}$$

for $m = 1, 2$, where $\mathcal{L}^2 \llcorner U_{ij}$ denotes the restriction of the two-dimensional Lebesgue measure to pixel cell $U_{ij} = (ih + h, jh + h)$ for $0 \le i \le M - 1, 0 \le j \le N - 1$. This choice conveniently satisfies that $\mu^m_h \in L^\infty(\Omega)$ since it can be expressed as

$$\mu^m_h = \frac{1}{h^2} \sum_{ij} \mu^m_{ij} \mathbb{1}_{U_{ij}}.$$

We solve (i) on a cell-centered grid $\Omega^{\boxdot}_h = \left\{ (ih + \frac{1}{2}h, jh + \frac{1}{2}h) \;\middle|\; 0 \le i \le M - 1, 0 \le j \le N - 1 \right\}$ using a standard finite differences approximation of the Laplacian defined as

$$(-\Delta_h f)_{ij} = -\frac{1}{h^2}(f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1} - 4f_{ij})$$

where $f_{ij} = f(ih + \frac{1}{2}h, jh + \frac{1}{2}h)$ is computed at cell centers, and due to boundary conditions $f_{-1,j} = f_{M,j} = f_{i,-1} = f_{i,N} = 0$ for all $0 \leq i \leq M-1, 0 \leq j \leq N-1$. Hence the discrete Poisson equation reads

$$(-\Delta_h f)_{ij} = \frac{1}{h^2} \sum_{mn} (\mu^1_{mn} - \mu^2_{mn}) \mathbb{1}_{U_{mn}}(ih + \frac{1}{2}h, jh + \frac{1}{2}h) = \frac{1}{h^2}(\mu^1_{ij} - \mu^2_{ij}).$$

The last step is to discretize the gradient operator, which is again done using forward differences with Neumann boundary conditions. Although f is defined on $\Omega_h^{\boxdot}$ the discrete gradient coincides with operator A as defined in (4.8). In summary the data term for H^{-1} noise removal reads as follows.

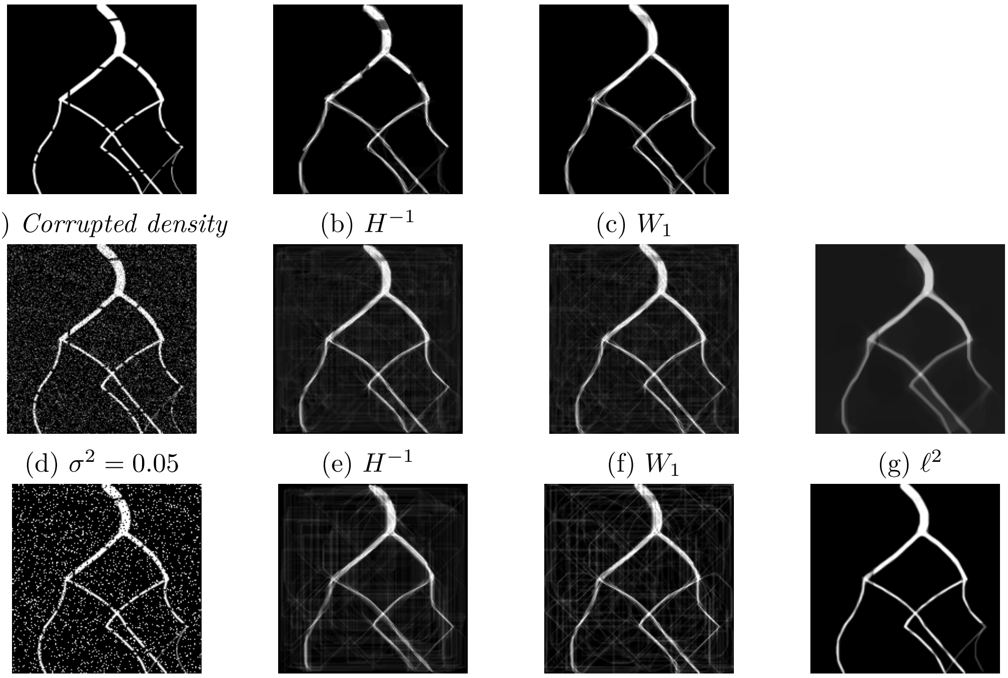
Given an input density $\mathcal{I} \in \mathbb{R}^{MN}$, defined by its linear coefficients according to (6.2), the associated discrete data term $G_{H^{-1}} : \mathbb{R}^{MN} \to [0, \infty)$ is

$$G_{H^{-1}}(Pa) = \|A(-\Delta_h)^{-1}(\mathcal{I} - Pa)\|_2^2,$$

which can informally also be written as

$$\begin{aligned} G_{H^{-1}}(Pa) = \min_{f,g} \quad & \|g\|_2^2 \\ \text{subject to} \quad & -\Delta_h f = Pa - \mathcal{I}, \\ & g = Af, \end{aligned}$$

to reflect that the operator $A(-\Delta_h)^{-1}$ applied in two steps using auxiliary variables, $f \in \mathbb{R}^{MN}$ and $g \in \mathbb{R}^{2MN}$.



(a) *Corrupted density* (b) H^{-1} (c) W_1

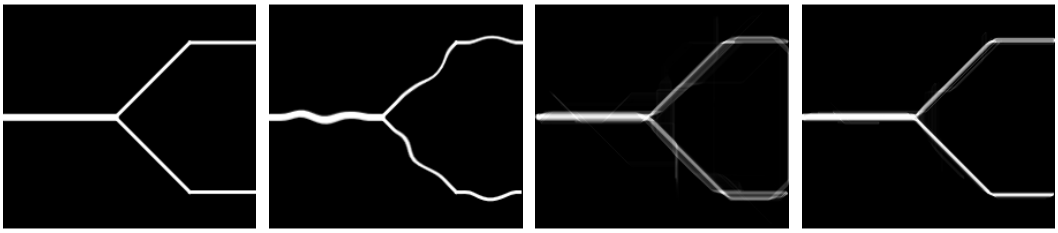
(d) $\sigma^2 = 0.05$ (e) H^{-1} (f) W_1 (g) ℓ^2

(h) *S&P 15%* (i) H^{-1} (j) W_1 (k) ℓ^1

Figure 6.7: *Restoration of corrupted and noisy densities with strong curvature regularization using different data fidelity measures. For* H^{-1} *and* W_1 *parameters were set to* $\alpha = 0.1$, $\beta = 5$, $\lambda = 0.25$ *and for* ℓ^p *noise removal as in section 3.7.1 the tuning parameter was set to* $\lambda = 0.75$.

Both the H^{-1} and the W_1 data fidelity terms reduced noise reasonably well and 'fused' the disconnected parts together. Due to the strong regularization both functionals tend to 'hallucinate' structures in the background. As speculated in the first chapter, salt and pepper noise introduces even stronger background noise.

While ℓ^2 based Gaussian noise removal from example 3.7.1 does completely eliminate the noise it also produces blurry line segments and tends to remove the thin segments at the bottom. The ℓ^1-data fidelity measure on the other hand is clearly better suited for removing salt and pepper noise than both new functionals. The H^{-1} data term can possibly be further refined by applying different boundary conditions.



(a) *Original density* (b) *Deformed* (c) H^{-1} (d) W_1

Figure 6.8: *Straightening of deformed density. Parameters* $K = 8$, $\alpha = 0.1$, $\beta = 50$, $\lambda = 0.5$.

When comparing the noise removal examples the differences between the Wasserstein-based and the H^{-1} data fidelity measures are not as obvious. Figure 6.8 shows a geometrically deformed density and the results of applying very strong regularization to recover the original density. The result using the H^{-1} distance is blurry and fails to compress the density back to the original thin lines. The Wasserstein distance on the other hand manages to recover the density almost perfectly. Since the W_1 fidelity measure corresponds to the minimal transportation costs and is conserving mass it might be better suited for recovering geometric transformations than for noise removal.

6.2 Comparison of discretizations

This section will focus on differences between the discretization using line measures and the one initially proposed in [7], which uses finite difference approximations. As said before the most critical part is the discretization of the directional derivative of the lifted quantity. In the initial article, the authors selected K uniformly distributed points on the S^1 and rescaled them as in Figure 6.9. Then to obtain the directional derivative of the lifted quantity μ they simply computed the difference between the measure 'values' at the centering node and the node with the direction in question, compare[7, Section 5.1] for further details.

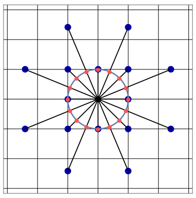
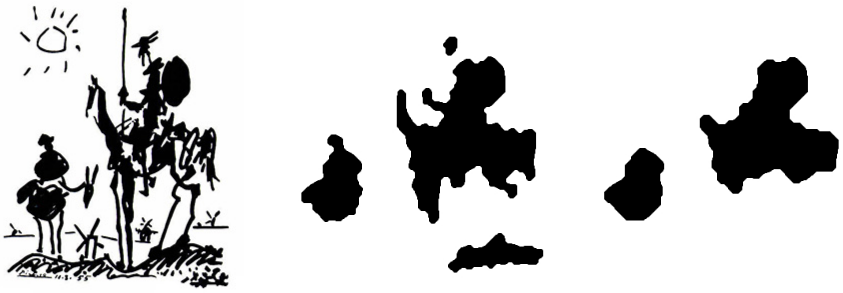


Figure 6.9: *Non-local finite difference scheme used to approximate the directional derivative.*

As the illustration already shows, not all nodes are aligned to the grid and hence to obtain their value a linear interpolation is necessary. To compare the two discretizations we selected one particular example from the original paper that showed the effects of varying the tuning parameter λ, which controls the weighting between regularization and the data term.



(a) *Original image for the examples.*

(b) TV, $\lambda = 0.5$

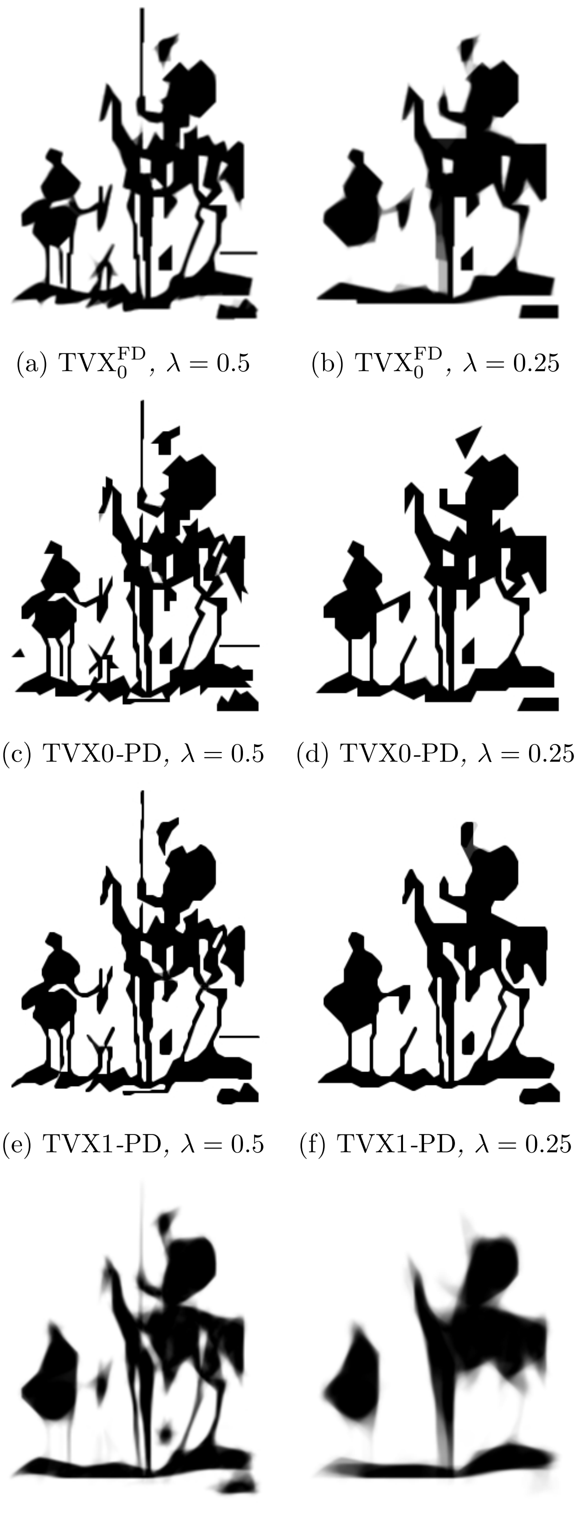
(c) TV, $\lambda = 0.25$

Figure 6.10: *Results of* TV*-regularization for varying* λ

Because this example contains many thin, elongated structures, applying pure length-based regularization leads to undesirable object shrinkage. Especially for strong regularization all lines are eliminated. While this example is well suited for studying the effects of varying the regularization weights, it also accurately shows the differences between the discretization approaches.

Due to the simple black and white image a simple linear segmentation force was used

$$f_{ij} = 0.5 - \mathcal{I}_{ij}.$$



(a) TVX_0^{FD}, $\lambda = 0.5$ (b) TVX_0^{FD}, $\lambda = 0.25$

(c) TVX0-PD, $\lambda = 0.5$ (d) TVX0-PD, $\lambda = 0.25$

(e) TVX1-PD, $\lambda = 0.5$ (f) TVX1-PD, $\lambda = 0.25$

(g) TVX_1^{FD}, $\lambda = 0.5$ (h) TVX_1^{FD}, $\lambda = 0.25$

Figure 6.11: *Comparison of discretization using a finite differences approximation and discretization using line measures. All images were computed using* $K = 32$ *discrete orientations,* $\alpha = 0.1$ *and* $\beta = 1$.

We denote by TVX_p^{FD} the regularization using the finite differences approximation as proposed in [7]. Figure 6.11 displays the result of binary-segmentation as in Section 3.7.2 using different data term weights λ.

The first notable thing is that the proposed discretization results in sharper images and is, with only a few exceptions, indeed binary. Even for strong regularization the solution preserves elongated structures well while effectively simplifying the results according to the prescribed functionals. The original discretization on the other hand yields non-binary and blurry results. The authors claimed that this behavior occurs due to non-uniqueness of the minimizer and less tightness of the relaxation. It could also be the case that this effect is amplified through the interpolation in the numerical discretization. Moreover to stop the iterations of the primal-dual algorithm the authors used a simple criterion based on the quantities $\left\|x^l - x^{l-1}\right\|$ and $\left\|y^l - y^{l-1}\right\|$ as in (5.2). While computing the results we modified the algorithm to save intermediate solutions, which for many iterations contained gray-shaded non-binary areas. And although the described quantities were notably small the shaded areas were eventually resolved after a very high number of iterations. Possibly the solutions of TVX^{FD} could be further refined with more iterations.

Again as noticed before TVX_0 leads to simplified polygonal shapes and TVX_1 to smoother shapes with fewer concave areas. In comparison to simple length-based regularization, both functionals return superior results that preserve the general structures well, while simultaneously simplifying the geometry with stronger regularization.

6.3 Performance evaluations

One topic of particular interest was to evaluate differences between the first-order primal-dual algorithm from [9] and a generic solver for linear and quadratic programming from the point of practical usability and performance.

While the primal-dual algorithm is very attractive because of its simple structure and easy implementation we were interested in finding out how such a naive implementation compares to a commercial solver.

6.3.1 Implementation details

Before we can begin with the performance comparisons and evaluations we will briefly summarize the setup used to compute the numerical results.

As a standard solver we chose MOSEK, which can among others solve large-scale linear, quadratic and second-order conic constrained optimization problems [4]. It provides a Matlab interface with replacements for common optimization toolbox functions, which made the integration into the Matlab source code very unobtrusive. The optimization model is built entirely inside Matlab and then passed on to MOSEK for solving.

The implementation of the first-order primal-dual algorithm is done in Julia [5], a relatively young high-level dynamic language that was designed for numerical and scientific computing. It provides access to many standard libraries, including good support for sparse matrix operations, and has a very similar syntax and semantic to Matlab.

The leading thought behind choosing a different environment than Matlab, was that the primal-dual algorithm approximates solutions through many iterations. And while Matlab performs well for vectorized expressions and provides many high-performing routines, we expect to gain a speed up by selecting a language that more efficiently handles control flows, such as loops, and provides better control over memory allocations. The first idea was to write a C++ program, which would have required to completely rewrite the existing Matlab code with very little resemblance. Additionally such an implementation introduces a lot of technical complexity we preferably wanted to avoid. While searching for alternatives we noticed that the developers of Julia claim through different benchmarks, that the language performs very similar to native C++ applications. Since porting the existing code to Julia required only little work and the results were promising we continued using it for this thesis.

For comparison and to support the choice of another environment, running time measurements for a Matlab implementation are also included.

If not otherwise stated, all computations in the following sections were run on a system with 12 GB physical memory and an Intel i5-4210U processor.

6.3.2 Memory consumption

The first factor of interest is memory consumption. We ran tests for different image sizes and numbers of discrete orientations to solve a binary-segmentation problem using TVX0-LP, TVX1-LP, TVX0-PD and TVX1-PD, and recorded the memory requirements while solving the optimization problems. While this is not an extensive study it provides a general view on the requirements and limitations of the different algorithms.

In all cases the memory required to build the models is not included. To achieve this all optimization models were generated in advance and stored using a compatible file format. For each of these files TVX-LP in MOSEK was then executed through a command line utility and the maximum required memory was measured. The same method was applied for TVX-PD in Julia. In that sense the data reflects the memory requirement to solve a given model, and does not contain any overhead introduced through Matlab. In all computations using MOSEK we applied the interior-point solver.

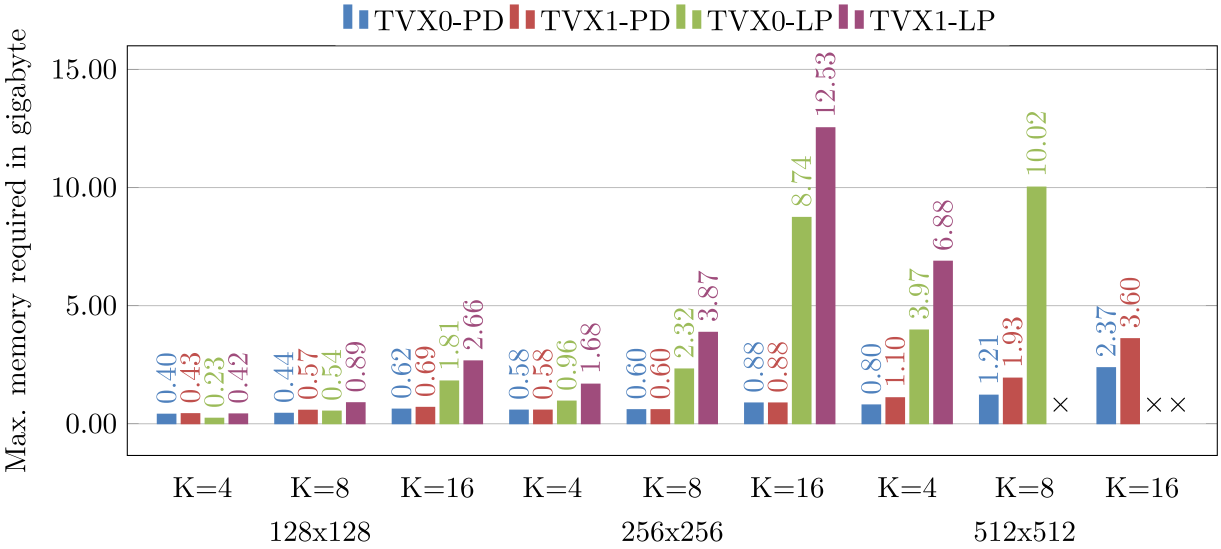


Figure 6.12: *Memory consumption measured for different image sizes and numbers of discrete orientations.*

The recorded data clearly shows that solving linear programs with MOSEK requires far more memory than the primal-dual counterpart. While the requirements for images of size 128×128 and $K = 4$ or $K = 8$ for the two methods are almost identical, increasing the number of discrete orientations to $K = 16$ results in a strong increase for TVX-LP using MOSEK. For images of size 512×512 we were unable to test $K = 8$ for TVX1-LP, and for $K = 16$ both TVX0-LP and TVX1-LP failed due to memory limitations.

Since our test system only had 12 GB of physical memory available computing problems using TVX1-LP is restricted to small images or few directions. With enough time we were still able to obtain results with TVX1-LP for an image size of 256×256 and $K = 16$ directions, however, while evaluating timings these settings cannot provide accurate results as much of the computational time is spent on swapping memory.

In general it can be observed that increasing the number of discrete orientations leads to a high jump in memory requirements for the LP. Results from the previous section have shown that for $K \leq 8$ strong directional preferences are still clearly visible and hence reducing the number of orientations is not a favorable option.

In conclusion, to apply TVX-LP using MOSEK a large amount of physical memory is required even for medium sized images. Although results in general become better with a larger number of discrete orientations, testing TVX-LP with $K = 32$ directions is challenging at best. In all tests TVX-PD required far less memory and scaled predictably with increasing image sizes and numbers of discrete directions, making it a favorable choice when only limited memory is available.

6.3.3 Running times

The following will focus on running times for the two different algorithms. As pointed out in the last section, evaluating running times for larger images is not possible with our test system, because MOSEK required more than the available physical memory and hence the results would be biased due to memory paging. It is however of interest how the two methods scale with different image sizes. On the test system we computed the problems for two different sizes, 64 × 64 and 128 × 128. Further we measured results for both regularization functionals TVX_0 and TVX_1 with the number of discrete orientations set to $K = 16$. Additionally we included results for images of size 256 × 256, which were partially computed on a high-performance cluster (ZIVHPC) with 32 GB of physical memory and two Quad Core AMD Opteron 2352 CPUs. These results have been marked, since they are not entirely comparable due to the different setup.

The running times can only be compared fairly if the solutions are computed to a similar accuracy. As in general the solutions to the optimization problems are non-unique, a simple comparison of the results from TVX-LP/QP/CP and the primal-dual algorithm is not possible. We selected the following optimality criteria.

TVX-LP/QP/CP For MOSEK three parameters control if a solution is optimal.

1. The primal feasibility error (PFEAS) needs to be below a threshold set to 10^{-8}.
2. The dual feasibility error needs to be below a threshold set to 10^{-8}.
3. The relative duality gap needs to be below a threshold set to 10^{-7}.

For further details, compare [4, 10.2.2.1 Interior-point termination criterion]. With these settings the results in general have a very high accuracy.

TVX-PD In the primal-dual algorithm an error-estimator is applied to decide if a solution is optimal. To make a fair comparison between the two algorithms it is necessary to choose an estimator that is based on the quality of the solution rather than the speed of convergence measured as change in the primal or dual variable. The second termination criterion from (5.3) measures both the convergence speed and the primal and dual feasibility error. The error tolerances were set to

- $\mathrm{tol}_1 = 10^{-7}$ for err_p, the estimated error in the primal variable,
- $\mathrm{tol}_2 = 10^{-6}$ for err_d, the estimated error in the dual variable ,
- $\mathrm{tol}_3 = 10^{-6}$ for err_pf, the primal feasibility error,
- $\mathrm{tol}_4 = 10^{-6}$ for err_df, the dual feasibility error.

To obtain a broader overview of different problems we selected the following examples.

1. **Binary segmentation** Provides a linear objective function.
2. **Gaussian noise removal** Provides a quadratic objective function.
3. **Inpainting** Consists of a set of equality constraints.

In all examples the regularization parameters were set to $\alpha = 0.1$, $\beta = 1$, $\lambda = 1$. As for the memory consumption the time for generating the optimization model is excluded and the results only show times required for solving a given model. Although in practice this step never required more than a few seconds, it could easily be eliminated by caching models for different image sizes and problems.

Problem	**Size**	**Reg.**	**Running time (s)**					
			MOSEK	PD-Julia		PD-Matlab		
			sec	sec	QM	sec	QM	QJ
Binary segmentation	64	TVX$_0$	24	27	1.13	36	1.50	1.33
		TVX$_1$	49	39	0.80	62	1.27	1.59
	128	TVX$_0$	188	90	0.48	158	0.84	1.76
		TVX$_1$	347	304	0.88	547	1.58	1.80
	256	TVX$_0$	*1602	965	*0.60	1772	*1.11	1.84
		TVX$_1$	*3310	1690	*0.51	2902	*0.88	1.72
Inpainting	64	TVX$_0$	35	475	13.57	535	15.29	1.13
		TVX$_1$	89	526	5.91	756	8.49	1.44
	128	TVX$_0$	441	1408	3.19	2211	5.01	1.57
		TVX$_1$	991	2836	2.86	4078	4.12	1.44
	256	TVX$_0$	*7751	5470	*0.71	9365	*1.21	1.71
		TVX$_1$	**25877	9853	–	18197	–	1.85
Gaussian noise removal	64	TVX$_0$	47	62	1.32	80	1.70	1.29
		TVX$_1$	149	305	2.05	497	3.34	1.63
	128	TVX$_0$	551	381	0.69	560	1.02	1.47
		TVX$_1$	1736	1554	0.90	2489	1.43	1.60
	256	TVX$_0$	*11114	2293	*0.21	3792	*0.34	1.65
		TVX$_1$	*17241	4488	*0.26	7953	*0.46	1.77

Table 6.1: *Measured running times in seconds. The values in column QM correspond to quotient (PD-* sec)/(MOSEK sec) and hence are greater than one if MOSEK was faster and less than one if the primal-dual algorithm was faster. Similarly the values in column QJ correspond to the quotient (PD-Matlab sec)/(PD-Julia sec) thus indicating for values greater than one that the implementation using Matlab was slower. Values marked with * have been obtained or computed from measurements on the high-performance cluster. Values marked with ** did not convergence within the maximum number of iterations.*

The first notable thing is that choosing a different environment than Matlab to implement the primal-dual algorithm is beneficial. Although we did not utilize parallelism or any special optimizations the approach using Julia is faster than the Matlab implementation in all of the example problems.

When comparing the running times of PD-Julia to the MOSEK implementation the results are not as clear and due to the small samples not very representative. For small and medium sized inpainting problems MOSEK required far less time for solving, presumably because many of the variables can be eliminated prior to running the actual minimization. In the other examples MOSEK returned the results quickly for the small image size, for the larger sizes PD-Julia was faster. As expected the running times clearly depend on the image problem in consideration.

In many samples the Matlab implementation of the primal-dual algorithm is the slowest approach. So when only PD-Matlab and MOSEK are available, MOSEK might be the better choice, keeping in mind that MOSEK possibly loses its performance advantage for larger problems. Moreover the computations with MOSEK on the high-performance cluster were not as successful as expected. Gaussian noise removal required a very long time and inpainting with TVX_1 failed to achieve the prescribed accuracy after more than seven hours. By looking at the output of the algorithm we observed that the convergence speed was extremely low. This could potentially be improved by instead solving an equivalent problem, for example by applying duality. But we did not further investigate these options.

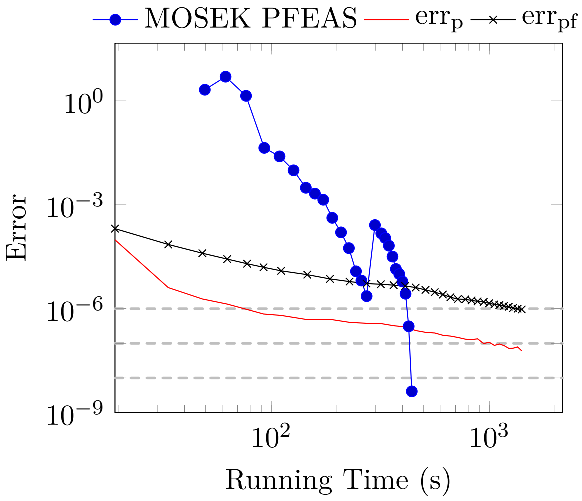


Figure 6.13: *Primal errors for MOSEK and PD-Julia measured over time for the inpainting problem using* TVX_0 *and an image size of* 128×128*. The horizontal lines indicate the different error tolerances.*

Figure 6.13 shows the different primal errors recorded while gathering the measurements for inpainting with TVX_0-regularization. The running times clearly show that MOSEK was significantly faster then the primal-dual approach. But the illustration of the errors shows a few more details. Firstly, obtaining a solution with medium accuracy can be achieved very quickly with the primal-dual algorithm, but the refinement to high accuracy takes many iterations. Secondly, before MOSEK starts the actual minimization it has a pre-solve and pre-optimization phase that already requires a long time and explains the gap at the beginning. Furthermore until the solution reaches medium accuracy already a large portion of the total solving time has passed.

In this example one can observe that bounding the primal-feasibility error below a threshold indeed serves as a restriction that can be stronger than bounding the root-mean squared error in the primal variable.

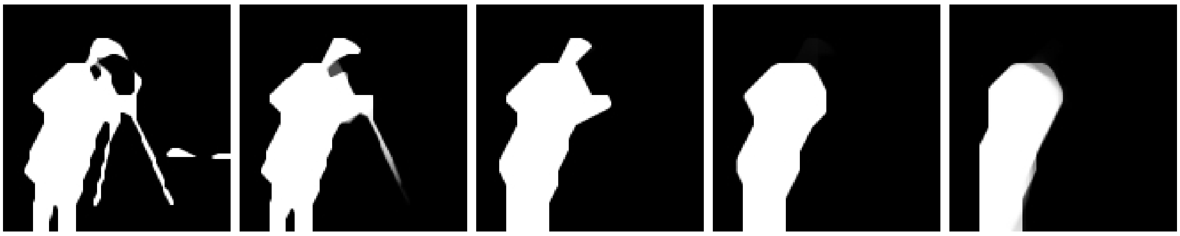
6.3.4 Performance conclusions

While writing this thesis we spent a large amount of time computing the different example problems for different test images and parameters. During these tests we mostly used smaller images with fewer discrete directions. For these types of problems MOSEK as solver was a good choice since it returned solutions with very high accuracy in a reasonable amount of time. With growing image sizes and more discrete directions MOSEK became slower and has two particular disadvantages over the primal-dual algorithm.

1. Reducing the required accuracy does not result in a strong speed-up. Before MOSEK starts the actual minimization it has a pre-solve and pre-optimization phase that requires a fixed amount of time for similar parameters, which strongly grows with the image size.
2. MOSEK's interior-point solver is unable to warm-start and hence refining a low-accuracy solution is not possible without entirely recomputing it.

The primal-dual algorithm on the other hand requires almost no preparation time and can compute approximate solutions that already after a few iterations reflect the selected parameters well enough to decide if refining the solution to high accuracy is worthwhile. Additionally it can be warm-started from a low-accuracy solution.

With both methods computing solutions to high accuracy can take a few hours depending on the problem structure and size. When the optimal parameters for the regularization are unknown and need to be tested, obtaining intermediate solutions can be helpful.



(a) $\lambda = 2$ (b) $\lambda = 1$ (c) $\lambda = 0.5$ (d) $\lambda = 0.25$ (e) $\lambda = 0.125$

Figure 6.14: *Low accuracy results of* TVX1-PD *binary segmentation after 1000 iterations for different regularization parameters. Running times for each image was 35 seconds with an image size of* 128×128 *and parameters* $\alpha = 0.1$, $\beta = 1$.

While the results in Figure 6.14 still contain low accuracy artifacts the effect of changing the regularization weight is already clearly visible. When computing the same problem using TVX1-LP with MOSEK, the pre-optimization phase already took roughly 70 seconds and the results from the first few iterations are practically unusable. Notice that computing the high-accuracy solution with TVX1-PD required almost 9-times as long, compare Table 6.1. For testing purposes we additionally increased the image size for the same problem to 256×256. MOSEK's pre-optimization phase then took 400 seconds, while running 1000 iterations to obtain a low-accuracy solution required only 107 seconds with TVX1-PD.

Moreover MOSEK does not provide a method for obtaining intermediate solutions, for example after every iteration, and thus the only way to test different parameters is essentially to each compute the full optimal solution.

In summary, for applications where the regularization parameters need to be adjusted before computing high accuracy results, using the primal-dual algorithm is a favorable choice.

In the previous sections we already pointed out that we experienced difficulties when using MOSEK to solve problems with higher numbers of discrete directions and larger image sizes such as 256×256 which from practical considerations is still very small. The main issue seems to be the large set of additional constraints introduced with increasing numbers of discrete orientations, as solving a similar sized problem with simple TV-regularization can be done very efficiently. We only applied MOSEK's interior-point solver, although a simplex solver is also available. While in preliminary tests the simplex solver required much less memory, it also showed significantly higher running times. Additionally it is restricted to linear programs limiting the possible applications.

7 Conclusions

In this thesis we first summarized the main concepts for a regularization framework based on curvature as proposed in [7]. The essential idea of that article was to consider a class of functionals acting on polygons that, depending on a metric on S^1, are able to compute a penalty by summing over all vertices and measuring the distances between incoming and outgoing directions. As first exemplary metric on S^1 they considered the discrete metric, which results in a functional that counts the number of vertices with distinct incoming and outgoing directions. Secondly they utilized the geodesic metric, resulting in a functional that measures the sum of the absolute external angles, which in some cases corresponds to the total absolute curvature. The authors showed that these functionals can be represented as a sum of line integrals along the polygon edges. In a first step this concept was then generalized to characteristic functions of polygons by introducing the concept of functional lifting of the gradient, and then afterwards applied to the sublevel sets of general images. To enable global optimization the authors then introduced a convex relaxation.

Starting from the observation that the functional lifting is essentially composed of line measures with support on only one-dimensional subsets, a new discretization has been proposed. This new discretization represents the lifted quantity by a sum of line measures with support on short line segments. By inserting these discrete measures into the regularization framework the functionals reduce to linearly constrained convex optimization problems that enable global optimization through a wide range of algorithms. Two possible methods have then been proposed. Firstly a preconditioned first-order primal-dual algorithm, and secondly a standard linear or quadratic programming approach.

Through different applications we were able to verify that the new discretization yields accurate results with sharp object boundaries. A comparison with the initial discretization, which is based on a finite differences approximation, showed that our approach is much better suited, most visibly when applying strong regularization. Especially for binary segmentation it can be observed that the initial method produces blurry non-binary regions that are presumably introduced by interpolation, a step that is not required by our approach.

Additional to the basic imaging problems we studied a specialized type of noise removal. While the results showed that the applied method indeed removes noise and preserves thin line-structures well, it can potentially be further enhanced by experimenting with different distance measures. Especially for impulse noise it was necessary to use very strong regularization to remove the particles, which had the side-effect that the functionals generated new structures in the background.

Finally we focused on the different methods for numerically solving the optimization problems. The approach using the primal-dual algorithm was implemented in a relatively young programming language named Julia and for comparison in Matlab. The linear or quadratic programs were solved using a commercial solver named MOSEK, which we were able to use freely through an academic license. While MOSEK computes solutions to a very high accuracy and was well suited for optimizing small problems, the high accuracy comes at a price of long running times for larger images. The implementations with the primal-dual algorithm on the other hand were fast in computing solutions with reduced accuracy, but required many iterations to reach a similar precision as MOSEK. But aside from the mathematical error, the optical differences between medium and high accuracy solution were usually almost imperceptible. In general the implementations demonstrate that curvature regularization with the proposed method can be applied very efficiently.

Since this thesis and the original article have demonstrated that the curvature regularization framework is well suited for many low vision problem, a next step could be to study more complex tasks. The example of binary segmentation for instance could be generalized to multi-label segmentation. Another possible direction of future study is to further utilize the special structure of the discretization using line-measures. One strategy for object recognition in computer vision is to apply some form of edge detection and then use this boundary description to determine similarity to objects from a set of templates. Hence it is crucial to accurately capture boundaries in a way that they are continuous and condensed to thin lines, which the curvature framework with the proposed discretization can naturally describe.

List of Figures

2.1 Comparison of different noise profiles 5

2.2 Relative interior for two-dimensional unit disk in three-dimensional space 6

3.1 Lifting of a smooth curve 10

3.2 Example of gradient and normal orientations for the 1-norm. 11

3.3 Lifting applied on a simple polygonal binary image 12

3.4 Single vertex polygon 13

3.5 Geodesic distance on S^1. 15

3.6 Admissible and non-admissible polygons 16

3.7 Illustration of vertex-counting functional 17

3.8 Illustration of total absolute exterior angle measuring functional 18

3.9 Polygon with piecewise smooth boundary 18

3.10 Image with balanced flows, its associated graph and its level set representation. . 23

3.11 Illustration of a video sequence of cells 25

3.12 Video of moving cells as three-dimensional image 25

4.1 Utilized directions for discrete line measures 27

4.2 Comparison of image and measure grid 28

4.3 Illustration of hat-functions 28

4.4 Precomputed offsets and coefficients 30

4.5 Illustration of flows for the directional derivative 31

4.6 Illustration of flow-balancing condition 32

4.7 Distribution of mass along different directions 32

4.8 Linearized indexing scheme 33

4.9 Sparse pattern of Operator D 34

6.1 Results of Gaussian noise removal 48

6.2 Results of salt and pepper noise removal 49

6.3 Results of binary segmentation 50

6.4 Comparison of TVX0 and TVX1 for binary segmentation and strong regularization 51

6.5 Results of inpainting 52

6.6 Results of deconvolution 53

6.7 Results of noise removal for special thin line-structure 57

6.8 Comparison between Wasserstein and H-1 data terms 57

6.9 Original discretization scheme using non-local finite differences 58

6.10 Results of TV-regularization for comparison with old discretization 58

6.11 Results of comparison to original discretization 59
6.12 Memory consumption for MOSEK and primal-dual implementation 61
6.13 Primal errors measured for MOSEK and PD-Julia over running time 64
6.14 Low accuracy results of primal-dual algorithm 65

List of Tables

6.1 Measured running times for utilized algorithms 63

References

[1] F. Alizadeh and D. Goldfarb, *Second-order cone programming*, Mathematical Programming **95** (2003), no. 1, 3–51.

[2] Luigi Ambrosio, Nicola Fusco, and Diego Pallara, *Functions of bounded variation and free discontinuity problems*, vol. 254, Clarendon Press Oxford, 2000.

[3] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré, *Gradient flows: In metric spaces and in the space of probability measures (lectures in mathematics eth zurich)*, 2nd ed., Birkhaeuser Basel, 2005.

[4] MOSEK ApS, *The mosek optimization toolbox for matlab manual. version 7.1 (revision 36).*, 2015.

[5] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, *Julia: A Fresh Approach to Numerical Computing*, ArXiv e-prints (2014).

[6] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge University Press, 2004.

[7] Kristian Bredies, Thomas Pock, and Benedikt Wirth, *Convex relaxation of a class of vertex penalizing functionals*, Journal of Mathematical Imaging and Vision **47** (2013), no. 3, 278–302.

[8] Kristian Bredies, Thomas Pock, and Benedikt Wirth, *A convex, lower semicontinuous approximation of euler's elastica energy*, SIAM Journal on Mathematical Analysis **47** (2015), no. 1, 566–613.

[9] Antonin Chambolle and Thomas Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, Journal of Mathematical Imaging and Vision **40** (2011), no. 1, 120–145.

[10] Lawrence C. Evans, *Partial differential equations (graduate studies in mathematics, vol. 19)*, 4th, reprinted with corrections(2008) ed., American Mathematical Society, 1998.

[11] Thomas Pock and Antonin Chambolle, *Diagonal preconditioning for first order primal-dual algorithms in convex optimization*, Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 1762–1769.

[12] Ralph Tyrrell Rockafellar, *Convex analysis*, 2. printing ed., Princeton Univ. Press, Princeton, NJ, 1972.