Master's thesis

2021

Vegard Gjerde Buset

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Vegard Gjerde Buset

Parameter selection methods for variational image restoration

June 2021

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Parameter selection methods for variational image restoration

Vegard Gjerde Buset

Master of Science in Physics and Mathematics
Submission date: June 2021
Supervisor: Markus Grasmair

Norwegian University of Science and Technology
Department of Mathematical Sciences

Abstract

We investigate the discrepancy principle and the L-hypersurface method as automated parameter selection methods for image denoising problems. Choosing optimal regularization parameters is always challenging, and the challenge is even bigger for multi-parameter regularization. We experiment with total generalized variation, a multi-parameter regularization functional, and the single-parameter regularization functional total variation. Total generalized variation is known to be a good multi-parameter regularization functional for image denoising, as its reconstructions avoids the piecewise constant property known as 'staircasing', which is commonly observed in total variation reconstructions.

The numerical solvers we test are based on convex optimization theory and use a Chambolle-Pock primal-dual solver. Single-parameter choice experiments show that the discrepancy principle performs better than the L-curve method for a RGB-image denoising problem, using a total variation regularization functional. For high noise levels, the discrepancy principle makes an almost optimal parameter choice for total variation. The numerical results show that an automated balanced discrepancy principle with Broyden's method as numerical solver performs better than the existing parameter choice algorithms for single-parameter methods. For images with fewer details and surfaces with almost constant intensities, total variation remains a better method. Tests of the L-hypersurface method suggest difficulties with choosing good parameters for total generalized variation by the means of this method.

Samandrag

Vi undersøker diskrepansprinsippet og ein L-hyperflate-metode som automatiserte parametervalmetodar i støyfjerningsproblem for bilete. Å velje optimale regulariseringsparametrar er alltid utfordrande, og utfordringa er enda større for regulariseringsfunksjonar med fleire parametrar. Vi eksperimenterer med total generalisert variasjon, ei fleirparameterval-metode, og med einparametervalmetoden total variasjon. Total generalisert variasjon er kjent som ein god støyfjerningsmetode for bilete, sidan rekonstruksjonar med denne metoden ikkje har den stykkvis konstante eigenskapen kjent som "trappeeffekta", som er vanleg i rekonstruksjonar basert på total variasjon.

Dei numeriske løysarane som er testa er basert på konveks optimeringsteori og ein Chambolle-Pock primal-dual-løysar. Eksperiment med einparametermetodar viser at diskrepansprinsippet presterer betre enn L-kurvemetoden for eit støyfjerningsproblem for eit RGB-bilete, når ein bruker total varisjon som regulariseringsfunksjon. For høge støynivå gjer ein ved bruk av diskrepansprinsippet nær optimale parameterval for total variasjon. Dei numeriske resultata viser at ein automatisert versjon av det balanserte diskrepansprinsippet med Broyden si metode som numerisk løysar presterer betre enn eksisterande parametervalalgortimar for einparametermetodar. For bilete med færre detaljar og overflater med nesten konstante intensitetar, forblir total variasjon ein betre metode. Testar av L-hyperoverflate-metoden antyder at det er vanskeleg å velje gode parametrar for total generalisert variasjon med denne metoden.

Acknowledgements

Writing a master thesis is often a lonely task, and even more so this year, with a pandemic raging across the world. My gratitude towards those who have supported me during the last year is however also higher, by an order of magnitude.

First and foremost I am deeply thankful towards Professor Markus Grasmair, who has been my supervisor both for the project thesis and for this master thesis. His enthusiasm, advice, support and impressive knowledge have been a huge inspiration for me. Without it and him, the story of my master thesis would be a quite different story. Our regular meetings and his detailed and constructive feedback to my work have shaped this master thesis into what it now has become.

A big thank you is also directed towards Anette. We have spent quite some time together this year, more than we originally expected, due to different restrictions, quarantines and other reasons to stay at home. It has been a pleasure as always. Thank you for your support when I have been stuck at some mathematical problem, and your understanding when I get an idea for the code at strange hours.

I would also like to thank Hans Julian Halvorsen, with whom I have worked for most group projects since 2019. He has taught me plenty of useful coding tricks, and his effort and desire for better-looking plots are something I can only admire and try to learn from. We have been a good team for the last years!

My parents Hallgeir and Mette should also be mentioned, for giving me a stable, educational and wonderful childhood and youth. Hallgeir does also deserve, together with my former teacher Henrik Kirkegaard, an extra thank you for finding suitable mathematical challenges for me at a young age, when the Norwegian school system failed to do so.

Finally I would like to thank NTNUI Volleyball. I have put in a lot of effort and worked hard for this organization during the last five years. But the club has given me so much, in ways words may fail to describe. It has been painful to watch younger students unable to take part in student sports to compete and form friendships this due to the pandemic. One of my greatest hopes for the next years is that many more Trondheim-students will be able to get almost as much joy from NTNUI as I did and still do.

Contents

1 Introduction

To determine optimal parameters is often essential in order to obtain good results in numerical optimization. It is in many cases also a challenging process, where it is hard to determine which set of parameters is the best. If multiple parameters need to be determined, the dimensionality of the problem is increased, and the challenge grows in size. In this text we consider such parameter selection methods on inverse problems, more specifically image denoising problems.

An introduction to image denoising problems is given in Section 2. This section also introduces the model we use to generate noise images and to evaluate reconstructions with objective measures. Another problem that arises in optimization and image denoising is which properties of a solution should be penalized. That is, how should the cost function be defined? In this text, two different regularization functionals are discussed and experimented on. These are total variation and total generalized variaton. Total variation produces piecewise constant reconstructions, where differences between neighbouring pixels are undesired. Total generalized variation is to a greater extent preserving existing structures in the image, and also avoids the piecewise constant effect from total variation.

Another difference between the two regularization functionals is the number of regularization parameters. Total variation is a single-parameter method with only one regularization term, whereas total generalized variation is a multi-parameter method with two regularization terms. These methods and their properties are discussed in Sections 3 and 4 respectively.

We are considering convex optimization methods, and will depend heavily on results from convex analysis. A short introduction to convex analysis with some central definitions and theorems is given in Section 2.6. Specifically, we will discuss the notion of duality and its use in convex optimization. This leads to the development of a numerical algorithm called the Chambolle-Pock algorithm, that we use to regularize images. In Section 5 we present numerical denoising results for two single-parameter choice methods, the discrepancy principle and the L-curve method. The discrepancy principle requires prior knowledge about the expected noise level, and the parameter choice is accordance with that knowledge. The L-curve method, on the other hand, is purely heuristic. This method chooses parameters that balance the contributions of the regularization term and the data discrepancy term.

We then generalize these methods to cope with multi-parameter regularization. This generalization is performed in Section 6, where we also introduce and suggest two possible numerical approaches to solve the multi-parameter choice problem. The balanced discrepancy principle is a generalization of the discrepancy principle which also requires the different terms to contribute equally to the cost function. The L-hypersurface method balances all regularization terms and the data discrepancy term, still without prior knowledge of the noise level. The results from these numerical methods are then finally presented and discussed in Section 7.

2 Preliminaries

Noisy signals are a well-studied topic within many areas of engineering. Removing noisy elements, disturbances or patterns is essential to obtaining precise and informative signals. Mathematicians have also been concerned with problems of this type.

In mathematics, noise removal is a special case of what is known as an **inverse problem**. An inverse problem is a problem where one has a set of observations, and wants to determine what caused them, based on the observations themselves. In this project a particular type of inverse problem is investigated, namely image reconstruction. The aim is to reconstruct an image based on a (possibly only partially) corrupted measurement of the same image. That is, given a noisy measurement v, one wants to determine the original signal $u^\dagger$. The *true* solution will be unknown throughout the process, meaning that different inverse problem techniques will be applied in order to create reconstructions of the original image or signal.

The noise level is used to describe the size of the noise in a signal. In real-world problems, this noise level is often unknown. There exist noise removal methods which require knowledge of the noise level, and there exist methods which do not. In this text, methods of both types will be used and tested on an image denoising problem.

2.1 Basics of the image model

Images can be considered as discrete signals. On a given image, we have nm pixels, which are small squares on the image on which the colour intensity is constant. We refer to n and m as the dimensions of the image. Moreover, for colour images, pixel intensities are stored in three different RGB-channels. RGB stands for red, green and blue, and the value in each of these channels determines the intensity of that colour in that pixel. So each pixel is a small square at a fixed location of the image, with a numerical value for the intensity of each of the three RGB-channels. Moreover, the RGB model is additive. This means that if we have the maximum intensity in all three channels, the image will be all-white. On the other hand, if the intensity is 0 in all channels, the image will be all-black.

The bit depth of an image refers to how many bits are used to store the intensities of each pixel. It is common to use an 8-bit depth, meaning that the intensity could take $2^8 = 256$ different integer values, ranging from 0 to 255. In the numerical calculations performed and explained in this text, this will always be scaled down to the unit interval $[0, 1]$, on which optimization algorithms and parameter selection methods will be applied.

The data we will work on will be measurements of digital signals. We will assume that we have a measurement $v \in \mathbb{R}^{3nm}$. Here, n and m are the dimensions of the image, and $v_{ij,k}$ is the measured pixel intensity in position (i, j) for the RGB-channel k. This means that an image is considered as a discrete signal for all practical means. From this noisy measurement v, we will try to denoise the signal and to recreate the original, noise-free signal $u^\dagger$.

We assume that both the original image $u^\dagger$ and the measurement v are elements of the Hilbert spaces U and V respectively. In the case of RGB image processing, we have that $U = V = \mathbb{R}^{3mn}$. Although this is only a very special situation, the derived results will remain valid for data in general finite dimensional Hilbert spaces, and also be applicable in situations where the two spaces are not identical.

The model we are going to use assume that the measurement $v \in V$ is a linear combination of the original signal $u^\dagger \in U$ and a random vector $\sigma w \in V$ in which the components are identically and independently distributed according to a normal distribution with expected value 0 and standard deviation $\sigma > 0$. Throughout this text, it is this standard deviation σ that often will be refered to as the **noise level**. The process of adding this Gaussian distributed noise is equivalent to having a discrete signal with additive white Gaussian noise [GTV11]. Moreover, we assume that $A : U \to V$ is a linear operator that is applied to u before the measurement is performed. This could for example be some kind of systemic measuring error or an effect caused by a blur on the camera lens. The suggested expression for the input signal to the image reconstruction problem can then be summarized as

$$v = Au^\dagger + \sigma w \in \mathbb{R}^{3nm}. \tag{2.1}$$

Throughout this project, A will be assumed to be known. This is not necessarily the case or a very realistic scenario, but is a reasonable simplification for analyzing the problem and to investigate different parameter choice methods and regularization functionals. If we in particular let $A = I$ (the identity matrix), we obtain a pure denoising problem. The task at hand is then only to identify and remove the noise with standard deviation σ. With these assumptions fulfilled, the measured data v is written as a linear combination of the original signal and the noise in the following way:

$$v = u^\dagger + \sigma w \in \mathbb{R}^{3nm}. \tag{2.2}$$

We want to reconstruct the original image $u^\dagger$ to as great an extent as possible, based on the noisy measurement v. Note that the expected value $\mathbb{E}[v]$ of v is $u^\dagger$, as the expected value of a sum is linear and the noise w is Gaussian distributed with mean zero in each component. This is shown by the following trivial calculation:

$$\mathbb{E}\left[u^\dagger + \sigma w\right] = \mathbb{E}[u^\dagger] + \sigma\,\mathbb{E}[w] = \mathbb{E}[u^\dagger] + 0 = \mathbb{E}[u^\dagger].$$

We recall the definition of a **well-posed problem** [Bor17]. If U and V are Hilbert spaces, and $A : U \to V$ is a mapping between the two Hilbert spaces, then the equation $A(u) = v$ is said to be well-posed if the following conditions are met:

- The equation has a solution $\hat{u} \in U$ for every $v \in V$.
- This solution $\hat{u}$ is unique.
- The solution depends continuously on the input v.

If these conditions do not hold, we say that the equation is **ill-posed**. In the setting with image denoising, we will usually have an ill-posed problem due to non-existence of solutions. Numerical algorithms can produce solutions in $\mathbb{R}^{3mn}$ for the pixel intensities, but not all these solutions generate meaningful images. If one only considers realistic reconstructions, one is left with a small subset of $\mathbb{R}^{3mn}$ as possible and realistic solutions. Thus the image reconstruction problem is an ill-posed problem.

2.2 Test image

The choice of images for the numerical experiments in this project is highly conventional. The KODAK image data set is a set of 24 8-bit RGB-images, and was released in 1991 by the American company KODAK. These images have been used for many purposes within image processing since [And+13]. For most of the experiments in this project, KODAK image number 22 has been used as a test image. The motif of the image is of a red barn. partly hidden behind some trees, with a pond in front of the barn. This image contains both areas with almost constant RGB-values (the barn wall), and areas with high details and many edges (the leaves of the tree in front of the barn). The original image is presented in Figure 2.1, and extracted from an online collection of the KODAK images presented by Rich Franzen [Fra10c]. The barn image is 512×768 pixels, meaning that the pixel intensities in each channel can be considered as subset of $\mathbb{R}^{393216}$.

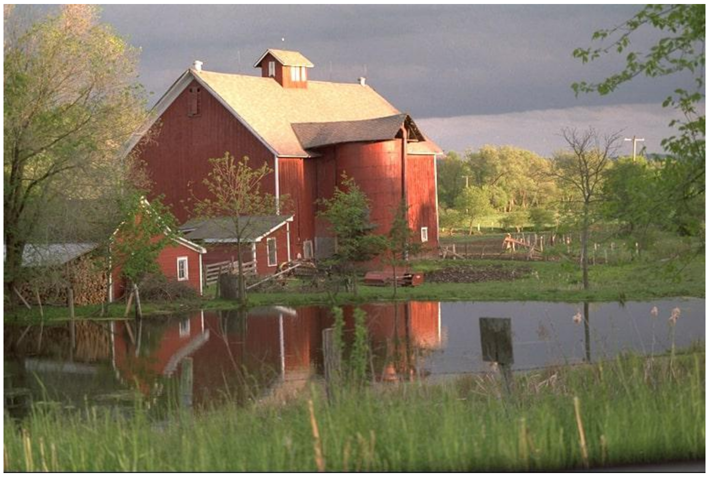


Figure 2.1: The orginal KODAK barn image. Photo: Cindy Branham.

2.3 Comparison of images

It will be difficult for the naked eye to distinguish between good reconstructions of images, so we want to find an objective measure for the quality of different reconstructions. It is thus important to establish and implement good measures to compare images, in order to evaluate and compare different reconstruction techniques and algorithms. Throughout this project many image reconstructions will be produced numerically, and there will be a need for comparing the quality of those reconstructions and thus also the methods and parameters that have been applied.

Moreover we should note that the comparison method we use requires knowledge about the original image, which serves as a *true solution.* In realistic inverse problems and image reconstructions this is of course not possible, as the true solution is unknown. For the purpose of testing methods and automating parameter choices however, knowing true solutions is a useful tool to identify good choices.

2.3.1 PSNR

PSNR, or peak-signal-to-noise-ratio, is a measure for how similar two discrete signals are. This can among other things be used to analyze and evaluate the quality of numerical reconstructions of the noisy and/or blurred images.

The first step of calculating PSNR of two images is to determine the mean square error (MSE) between the true solution u and the reconstructed solution $\hat{u}$. We can recall from for example [SA10, p 196-197] that the mean square error over a $m \times n$-image is given as

$$\text{MSE} = \sum_{k=1}^{3} \sum_{i=1}^{mn} \frac{(u_{ik} - \hat{u}_{ik})^2}{3mn}.$$

Here, k is the channel. This means that we find the mean square error of a vector of length $3mn$, where the image $u \in \mathbb{R}^{3mn}$ is an $n \times m$-dimensional RGB image. The other value that must be determined is the maximum pixel value over all three RGB channels for the noise-free image, denote this $u_{\max}$. Then, the PSNR value is given as

$$\text{PSNR} = 10 \log\Big(\frac{u_{\max}^2}{\text{MSE}}\Big). \tag{2.3}$$

Basic algebraic reformulations of the logarithmic expression in (2.3) give two identical formulations of PSNR.

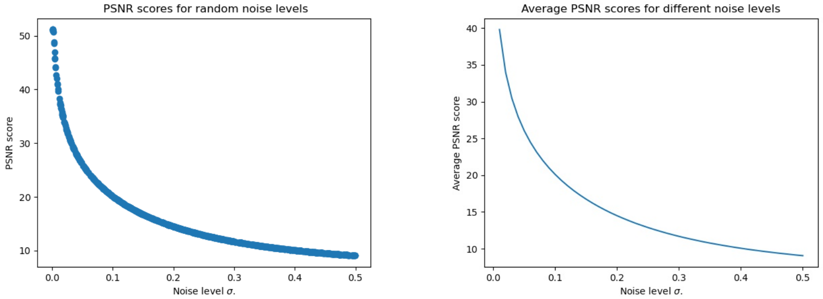
$$\text{PSNR} = 20 \log\Big(\frac{u_{\max}}{\sqrt{\text{MSE}}}\Big) = 20 \log(u_{\max}) - 10 \log(\text{MSE}).$$

From the definitions we observe that the lower the mean square error, the bigger the PSNR score. The PSNR score is also proportional to the logarithm of the largest pixel value in the original image, but this is a fixed value for a given image. Images with high PSNR-scores are thus more similar pixel-wise than images with lower PSNR-scores.

However, there are challenges related to the use of PSNR as a measure. PSNR compares the pixelwise intensity. This means that if the images are somewhat shifted or rotated, PSNR will not "recognise" the original image. Looking at each pixel separately, the measure will not take into account the smoothness of pixel transitions or the preservation of patterns in the image. This is a weakness with this measure.

In order to illustrate how the PSNR measure works in practice, we perform an experiment in which we use the KODAK image in Figure 2.1[Fra10c]. We generate 1000 random noise levels from a continuous and uniform distribution between 0.0 and 0.5, and use these to generate 1000 noisy images from the original one. The PSNR scores of the realizations are calculated, and then plotted against the noise levels. The results are presented in Figure 2.2a. We observe that the PSNR score decreases as the noise level increases, as one would expect. In Figure 2.2b, 50 uniformly spaced noise levels between 0 and 0.5 have been applied at the same image. For each noise level, ten different images have been produced with that noise level, and then the average PSNR value of those images are plotted against the noise level. We observe that this curve follows the shape of the one from Figure 2.2a.

Figure 2.2: Examples of PSNR:

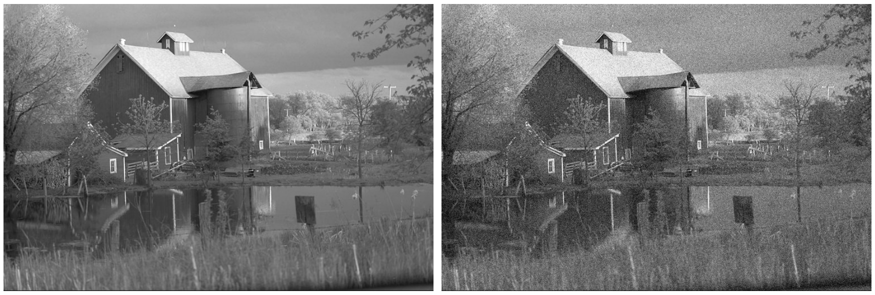


(a) PSNR scores for 1000 randomly generated noise levels.

(b) Average PSNR value for fixed noise levels.

2.3.2 SSIM

Figure 2.3: Example of greyscale images without and with noise.



(a) Greyscale original KODAK image

(b) Noisy greyscale image, $\sigma = 0.1$

A measure that does take preservation of structures into account is the socalled **structural similarity index measure**, or SSIM for short. This measure was suggested by [Wan+04], and divides the reconstruction evaluation into the comparison of three different properties, namely contrast, luminance and structure.

The PSNR measure that was introduced in the last subsection compares the images pixel by pixel. This means that an image that has been reconstructed accurately, except for a minor shift along one of the axes, may get a bad PSNR score. By a visual inspection, such a picture does look like a good reconstruction if it contains the structures of the original image. This is an effect that SSIM copes with, by combining and including the three different properties of the image, contrast, luminance and structure.

The SSIM measure is however not applicable directly on RGB images. We first need to export the image to a greyscale image, and compare greyscale images. These images are stored as Numpy arrays in $\mathbb{R}^{mn}$, which only contain one intensity value in each pixel. THE KODAK image can be seen as a greyscale image in Figure 2.3. The comparison between two images u and v in the SSIM metric is performed by first calculating

$$\mu_{uk} = \frac{1}{nm} \sum_{1}^{nm} u_i \text{ and } \mu_v = \frac{1}{nm} \sum_{1}^{nm} v_i$$

as the mean intensity of the respective images u and v. We continue by considering the standard deviation for the images. We define

$$\sigma_u = \Big(\frac{1}{nm-1} \sum_{1}^{nm} (u_i - \mu_u)^2 \Big)^{\frac{1}{2}}.$$

Similarly, we have that $\sigma_v = \Big(\frac{1}{nm-1} \sum_{1}^{nm} (v_i - \mu_v)^2 \Big)^{\frac{1}{2}}$. The values σ_u and σ_v are used for contrast comparison of the two images u and v.

Finally, the structure comparison is performed on normalized signals with respect to their standard deviations. That is the vectors $\frac{\mathbf{u}-\mu_u}{\sigma_u}$ and $\frac{\mathbf{v}-\mu_k}{\sigma_k}$, where the mean is subtracted from each component of the vector, before dividing by the standard deviation.

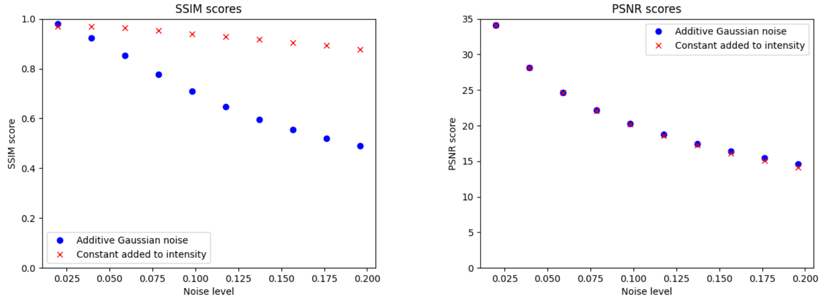
These are the main vectors and values needed to establish a SSIM measure. Functions for contrast, luminance and structure is combined, see [Wan+04] for details. After some simplifications the authors suggest the following formula to estimate the quality of the reconstruction:

$$\mathrm{SSIM}(u, v) = \frac{(2\mu_u \mu_v + C_1)(2\sigma_{uv} + C_2)}{(\mu_u^2 + \mu_v^2 + C_1)(\sigma_u^2 + \sigma_v^2 + C_2)}. \tag{2.4}$$

Here C_1 and C_2 are parameters with different purposes. C_1 is used to avoid unstability in pictures with low average pixel values [Wan+04], and is usually proportional to the bit depth squared. The proportionality constant is positive, and usually much smaller than 1. C_2 is also commonly chosen proportional to the square of the bit depth, again with a small proportionality constant[Wan+04]. In all SSIM calculations in this thesis, the built-in SSIM measure in the `scikit-image` package have been used, with the standard parameters and constants from that package. If two images are identical, they will get a value of 1.0 in the SSIM metric.

An example illustrating the properties of SSIM has been produced for a situation where the shortcomings of the PSNR metric are visible. A constant value K has been added to all pixel intensities in the barn image. Since we scale images down to the unit interval, this corresponds to a expected noise level of $\frac{K}{255}$. We can compare how the different metrics behave when analyzing this kind of noise, compared to the additive Gaussian noise we have considered earlier. Comparisons to the original image $u^\dagger$ are performed in both metrics. Since PSNR only is based on the pixel intensities, the score in this metric is drastically reduced. In the SSIM metric however, it is relevant that the structures of the original image and the image with shifted pixel intensities remain the same, and the SSIM score also remains relatively high. We have tested for ten different values of K, uniformly spaced between 5 and 50. For each value of K, we have also considered a realization of the corresponding noise level $\sigma \approx \frac{K}{255}$, and evaluated this realization in both metrics. The results are presented in Figure 2.4. From the plots we clearly see that the structural similarity index is much higher for the image with shifted intensities than for the image with Gaussian noise, as the key structures of the shifted image are preserved. In the PSNR metric however, the new approach leads to almost the same PSNR scores as the Gaussian distributed noise at the same noise level. This is a clear illustration of a situation in which the SSIM metric is a good one, as it identifies that the structures of the image are preserved after the shift.

Figure 2.4: SSIM and PSNR scores of images shifted by adding constant pixel intensities.



(a) SSIM

(b) PSNR

2.4 Regularization of inverse problems

A key element of solving ill-posed inverse problem, is to regularize the problem. This is done by creating or designing restrictions for the solutions, 'forcing' it to be an element of some smaller subspace that is easier to control and describe. This regularization will change the problem somehow, and the aim is to obtain an easier solvable problem, in such a way that the solution remains close to the solution of the original problem.

Regularization is commonly performed by introducing **regularization functionals** $R_i : U \to \mathbb{R}^+$. These functionals penalize different un-desired properties of the reconstructed solution u. If one applies Tikhonov regulariztion in an optimization problem, there is a total of k regularization functionals $R_i(u)$. Each of the k regularization functionals has a regularization parameter $\lambda_i > 0$, which is used to balance the different regularization functionals against each other. Depending on the choices of the different λ_i, different properties will be important when the optimal reconstruction is decided. The optimization problem one obtains when applying Tikhonov regularization can be written as

$$\min_u = L(u, v) + \sum_i^k \lambda_i R_i(u). \tag{2.5}$$

Here v is the original signal and u is the reconstruced signal. The different λ_is are the Tikhonov regularization parameters which take care of the **weighting** of the regularization terms. Higher values of λ lead to more regularized solutions, but possibly also greater deviations from the true solution. $L(u, v)$ is often referred to as the **data discrepancy** term, and may for example be a least squares term $\frac{1}{2}\|u - v\|_2^2$.

Exactly how regularization is performed, depends on which property one want to restrict. In image analysis, a commonly desired property is the **smoothness** of the image [Sch+09, p 116]. Penalizing sharp edges within the image is an example of a technique that copes with this requirement, and thus many regularization functionals contain the gradient of the image. [Sch+09, p 115-117] continues by defining a first order regularization functional as a regularization functional where the highest order of derivatives of the image is 1. In general, we can consider a k-order regularization functional as a regularization functional where the highest order of derivatives of the image is k.

2.5 The gradient operator D and quadratic regularization

A very basic and intuitive regularization functional is one that is based on **quadratic regularization**. We will briefly look into this regularization technique, and perform some simple numerical experiments. From these experiments we will visualize the consequences of **over-regularization** and also show results of image denoising after a search for the optimal regularization parameter for quadratic regularization.

Quadratic regularization uses the L_2-norm of the gradient Du as the regularization term. For a parameter $\lambda \in \mathbb{R}_+$, the functional that should be minimized is given as

$$S(\lambda)(v) = \operatorname*{arg\,min}_{u \in U} \frac{1}{2}\|Au - v\|_2^2 + \frac{\lambda}{2}\|Du\|_2^2. \tag{2.6}$$

In our analysis, the noisy measurement consists of RGB-values that are discrete integers between 0 and 255 in each, discrete pixel. The gradient operator will also be evaluated in the nm discrete pixels. Each RGB-channel is still treated separately, and for each channel the gradient consists of horizontal and vertical derivatives (x-direction and y-direction). Thus, since $u \in \mathbb{R}^{3mn}$, we have that $D : \mathbb{R}^{3mn} \to \mathbb{R}^{3mn \times 2}$. We will however scale the signals down to the unit interval. When we send scaled discrete signals to the different numerical algorithms, we have that $\text{Ran}(u) \subset [0,1]^3$. This means that the signal $u(x,y)$ in each channel is discrete and scaled to the unit interval, by dividing **all** pixel values with 255. This means for example that a pixel with a RGB-value of 153 in one of the channels, will have the RGB-value 0.6 in the same channel in the scaled image, regardless of what the maximum value (over the entire image) in that RGB-channel is.

The gradient operator D is an operator based on forward differences in two dimensions. Forward differences are discretized derivatives, and two-dimensional forward differences can thus be considered as discretized partial derivatives. We will observe that this operator is enough to ensure some regularization properties.

As we operate on two-dimensional grids and deal with two-dimensional digital signals, we need to determine forward differences in multiple directions. In each pixel, we want to find the forward difference in the x-direction, and the forward difference in the y-direction. We follow the convention given by Owren, and impose homogeneous boundary conditions at the edges where it not possible to calculate any forward difference [Owr17]. In practice, this is to impose homogeneous Neumann conditions on our gradient operator. The operators D_x and D_y in the different directions are defined in the following way.

$$D_x(u_{(i,j)}) = \begin{cases} u_{(i+1,j)} - u_{(i,j)} \text{ for } i < n \\ 0 \text{ for } i = n \end{cases} \tag{2.7}$$

$$D_y(u_{(i,j)}) = \begin{cases} u_{(i,j+1)} - u_{(i,j)} \text{ for } j < m \\ 0 \text{ for } i = m \end{cases} \tag{2.8}$$

Combining these definitions, we obtain the following formal definition of the gradient operator in each RGB-channel:

$$D(u_{(i,j)}) = \begin{cases} (u_{(i+1,j)} - u_{(i,j)}, u_{(i,j+1)} - u_{(i,j)}) \text{ for } i,j < n \\ (u_{(i+1,j)} - u_{(i,j)}, 0) \text{ for } i < n = j \\ (0, u_{(i,j+1)} - u_{(i,j)}) \text{ for } j < n = i \\ (0,0) \text{ for } i = j = n \end{cases} . \tag{2.9}$$

We denote $D_x(u_{(i,j)}) = D(u_{(i,j)})^{(1)}$ and $D_y(u_{(i,j)}) = D(u_{(i,j)})^{(2)}$.

Quadratic regularization is the regularization technique in which the only regularization functional is given by $\|Du\|_2^2$, as introduced in Equation (2.6). The minimization problem in a denoising setting with $A = I$ is given as

$$\mathrm{QR}_\lambda(v) = \operatorname*{arg\,min}_{u \in U} \Big[\frac{1}{2} \|u - v\|_2^2 + \frac{\lambda}{2} \|Du\|_2^2 \Big]. \tag{2.10}$$

We derive the first order optimality condition of this problem in order to develop a numerical algorithm to solve the minimization problem. The first order optimality condition in the case of denoising is

$$(u - v) + \lambda D^* D u = 0.$$

This is equivalent to $(I + \lambda D^* D)u = v$, and thus we have an equation of the form $Mu = v$ for some linear operator M. This is all we need to apply a simple numerical solver such as the conjugate gradient method. And we will now apply just that method, mainly to illustrate some properties and weaknesses with quadratic regularization. We use the conjugate gradient method, with the matrix $I + \lambda D^* D$ as the linear operator. For an introduction to the conjugate gradient method, the reader may consult Appendix A. This is not a very sophisticated numerical method, and is included in this text mainly to illustrate some properties of quadratic regularization on an actual image.

We need to verify that the matrix $(I + \lambda D^* D)$ is positive definite, in order to use it as the linear operator in the conjugate gradient method. This is a pretty straight-forward calculation. Positive definiteness for a matrix M is defined such that for all vectors $u \neq 0$, $u^T M u > 0$. We can split $u^T (I + \lambda D^* D) u$ into two terms $u^T I u$ and $\lambda u^T D^* D$. Then we can observe that

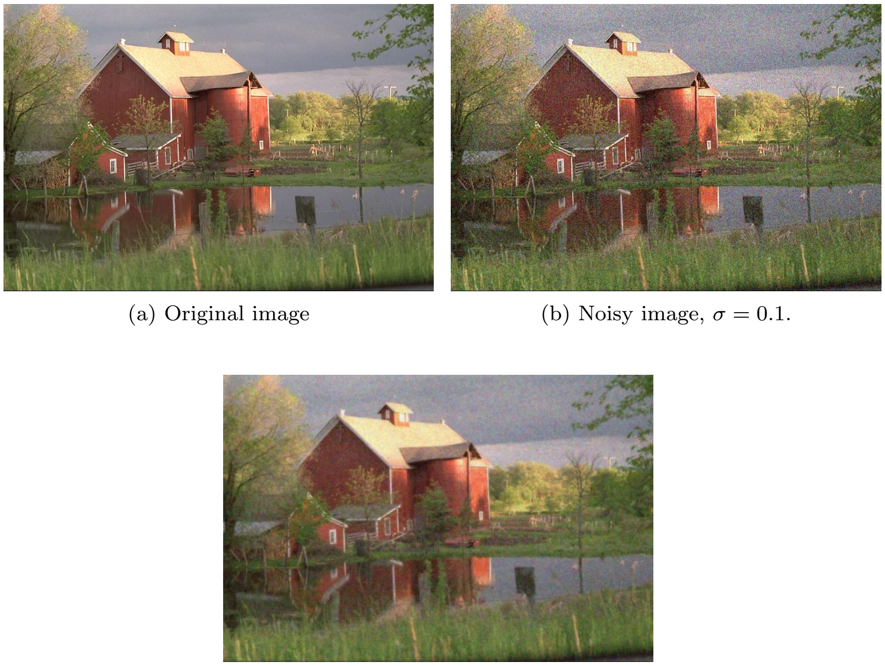
$$u^T (I + \lambda D^* D) u = \|u\|_2^2 + \lambda \|Du\|_2.$$

This is bigger than 0 for all $u \neq 0$, as we recall that $\lambda > 0$. Thus, $I + \lambda D^* D$ is positive definite. This result can also be derived by noting that we are solving a coercive and quadratic optimization problem for which $I + \lambda D^* D$ is the Hessian of the problem. Since this Hessian is positive, the matrix is positive definite.

The stability of the method is also of interest, as well as its ability to converge fast. The stability of a numerical method is often measured by the condition number κ, which measures how much the output is affected by minor changes or errors in the input. A common definition of this number is given by the fraction between the maximum and the minimum singular value. In our numerical tests, we restrict ourselves to tests for $\lambda \leq 5$ in order to avoid over-regularization. The fact that we control the regularization parameter gives us, for the matrix $I + \lambda D^* D$, that the singular values $\sigma_i \in [1, 1 + 4\lambda]$. In the worst case scenario we have that $\sigma_{\max} = 21$, and thus $\kappa \leq 21$. This is a low condition number, and we would expect the conjugate gradient approach to be a stable one. The low condition number and the small variance in singular values also cause the conjugate gradient algorithm to converge fast, without the need for a preconditioning algorithm.

The numerical results from the CG algorithm are shown in Figure 2.5. From the examples of denoised images, we observe that the algorithm is able to remove some of the noise from the images. However, the images become somewhat more blurry in the denoising process. This happens in particular if the regularization parameter λ is big, as the weighting of the gradient term then dominates, and the minimization algorithm focuses on removing small differences between neighbouring pixels.

Figure 2.5: Example of quadratic regularization denoising.



(a) Original image

(b) Noisy image, $\sigma = 0.1$.

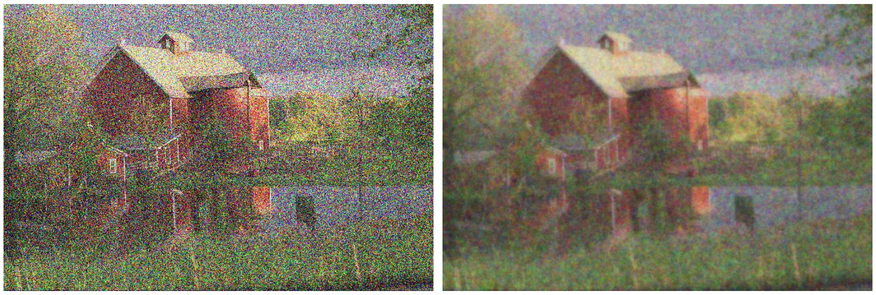
(c) Denoised image, $\lambda = 1.25$.

If we increase the noise level more, say $\sigma = 0.4$, the blurring effect of the quadratic regularization algorithm becomes very apparent, as illustrated in Figures 2.6.

2.6 Convex analysis

Many numerical results in this paper will depend on knowledge from convex analysis. The aim of this subsection is to give a short introduction to key concepts of

Figure 2.6: Example of the blurry effect of quadratic regularization with large noise level.



(a) Noisy image, $\sigma = 0.4$

(b) Image denoised with $\lambda = 5$.

convex analysis. Important definitions and theorems will be presented, as well as some proofs. A few examples, in which convex analysis is used, will also be presented. The objective is that this chapter will give the reader the knowledge and background from convex analysis needed to follow the arguments and calculations later in the paper. In particular, subdifferentials and sub-gradients, duality and proximal point mappings will be essential to derive optimality conditions. Convex conjugates will also be introduced, and will become essential at a later stage in the text. For the entirety of this subsection U is defined to be a finite-dimensional Hilbert space.

Convex analysis is concerned with convex functions on convex sets, and the properties of these functions. A natural starting point for convex analysis is thus to define convex sets and functions.

Definition 2.1. Convex sets.
A convex set C is a subset of a vector space, such that for any x_1, $x_2 \in C$, and for any $t \in [0, 1]$ we have that $tx_1 + (1-t)x_2 \in C$.

Definition 2.2. Convex functions.
A convex function $f(x)$ is an extended real-valued function from U to $\mathbb{R} \cup \{+\infty\}$ such that the following inequality holds for all u_1,$u_2 \in U$ and for all $t \in (0, 1)$:

$$f\Big(tu_1 + (1-t)u_2\Big) \leq tf(u_1) + (1-t)f(u_2).$$

A relevant property for convex functions in the finite dimensional case is that all such convex function are (locally Lipschitz) continuous on the interior of their domains [BL06, p 65].

Definition 2.3. Domain of a convex function
The domain of a convex function $f : U \to \mathbb{R} \cup \{+\infty\}$ is the set of all vectors u for which $f(u)$ is finite. The domain is denoted dom$f$ and formally defined as

$$\text{dom} f = \{u \in U : f(u) < \infty\}$$

After introducing the domain in Definiton 2.3, we are ready to define a **proper, convex function**. This is defined as a convex function with a non-empty domain. That is, a function $f : U \to \mathbb{R} \cup \{+\infty\}$ for which $f(u) < \infty$ for at least one $u \in U$, is proper. The set of proper and convex functions mapping from U that additionally are lower semi-continuous, is denoted by $\Gamma_0(U)$.

Another important concept that will be essential throughout the paper is the concept of convex conjugation.

Definition 2.4. Convex conjugates
Let U be a finite dimensional Hilbert space, as before. Then the (Fenchel) convex conjugate of a proper function $f : U \to \mathbb{R} \cup \{+\infty\}$ is $f^* : U \to \mathbb{R} \cup \{+\infty\}$ defined as:

$$f^*(p) = \sup_{u \in U} \Big[\langle u, p \rangle - f(u) \Big]$$

Definition 2.5. The biconjugate
The biconjugate of $f : U \to \mathbb{R} \cup \{+\infty\}$ is $f^{**} = (f^*)^*$.

Definition 2.6. Projection on a convex subset
Let $C \subset U$ be non-empty, convex and closed. The **projection** of a vector $u \in U$ onto C is then defined as the unique solution to

$$\min_{x \in C} \|x - u\|_2^2$$

The projection of the vector u on C is denoted $\pi_C(u)$.

Example - Conjugate of characteristic function

A simple example of a convex conjugate which will turn out useful, is the conjugate of the characteristic function $i_C(x)$ of an arbitrary convex set C. We assume C is a subset of U. Then the characteristic function $i_C : U \to \mathbb{R}$ is defined as

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

Then, from Definition 2.4 we get $i_C^*(p) = \sup_{u \in U} \Big[\langle u, p \rangle - i_C(u) \Big]$

Whenever, $u \notin C$, we have that $i_C(u) = +\infty$, and whenever $u \in C$ we have that $i_C(u) = 0$.The expression for the conjugate can then be simplified to

$$i_C^*(p) = \sup_{u \in C} \langle u, p \rangle,$$

which is exactly the support function of the set C [HL01, p 134].

2.6.1 Subdifferentials

Subdifferentials are useful whenever the convex functions one considers is non-differentiable.

Definition 2.7. Let $f \in \Gamma_0(U)$. Then the **convex subdifferentia**l of f at u is

$$\partial f(u) = \begin{cases} \{\tilde{u} \in U : f(v) \geq f(u) + \langle \tilde{u}, v - u \rangle \text{ for all } v \in U\}, & \text{if } u \in \mathrm{dom}(f), \\ \emptyset, & \text{if } u \notin \mathrm{dom}(f) \end{cases}.$$

Elements of sub-differentials are called **sub-gradients**.

A known result is that for a convex, lower semi-continuous function f, which is Gâteaux differentiable at a point $x \in U$, we have that $\partial f(x) = \{\nabla f(x)\}$. That is, the only element of the subdifferential coincides with the actual gradient of the differentiable function.

Theorem 2.1. $\hat{u} \in U$ is a minimizer of $f : U \to \mathbb{R}\{+\infty\}$ for $f \in \Gamma_0$ if and only if $0 \in \partial f(\hat{u})$.

Proof. Assume $\hat{u} \in U$ minimizes $f \in \Gamma_0$. By definition, for any arbitrary element $u \in U$, we have that

$$f(u) \geq f(\hat{u}). \tag{2.11}$$

Now, we can add the inner product between zero and $u - \hat{u}$, which leads to the inequality

$$f(u) + \langle 0, u - \hat{u} \rangle \geq f(\hat{u}) + \langle 0, u - \hat{u} \rangle.$$

Finally we note that this inner product is equal to zero, and remove it from the left hand side to obtain

$$f(u) \geq f(\hat{u}) + \langle 0, u - \hat{u} \rangle.$$

This is precisely the expression from the definition of the convex subdifferential above, since u is an arbitrary element in U. We have then shown that $0 \in \partial f(\hat{u})$ if $\hat{u} \in U$ is a minimizer of f. However, we can note that all the steps and transformations are valid in both directions, meaning that we can reverse the steps of the proof to obtain the 'only if'-result of the proof. Thus $\hat{u} \in U$ is a minimizer of $f : U \to \mathbb{R}\{+\infty\}$ for $f \in \Gamma_0$ if and only if $0 \in \partial f(\hat{u})$. □

This theorem gives us a useful result and a connection between minimizers of proper convex functions and sub-differentials. The result will be used to derive optimality conditions for the minimization problem.

2.6.2 Duality

We want to solve optimization problems of the form

$$\min_{u \in U} [f(u) + g(Bu)], \tag{2.12}$$

where $B : U \to V$ is a linear operator. This kind of optimization problems have solutions which are described by the Fenchel-Rockafeller theorem, and rely on the notion of duality. We will state the result of the theorem for general Hilbert spaces U and V.

Theorem 2.2. Fenchel-Rockafeller theorem
Let U and V be Hilbert spaces. Moreover, let $f \in \Gamma_0(U)$ and $g \in \Gamma_0(V)$. Let $B : U \to V$ be a linear operator. Now assume that the primal problem $\min_{u \in U}[f(u) + g(Bu)]$ admits a solution $\hat{u}$. Assume also that there exists an $\bar{u} \in \text{dom}(f)$ with $B\bar{u} \in \text{int}(\text{dom}(g))$. Then:

1. The dual problem

$$\min_{p \in V} [g^*(p) + f^*(-B^* p)]$$

admits a solution $p^* \in V^*$.

2.
$$\min_{u\in U}[f(u)+g(Bu)] = \max_{p\in V}[-g^*(p)-f^*(-B^*p)]$$

3. $\bar{u}$ and $\bar{p}^*$ are solutions to the primal and dual problem respectively if and only if $-L^*\bar{y}^* \in \partial f(\bar{x})$ and $\hat{y}^* \in \partial g(L\bar{x})$.

The proof of this theorem is long and somewhat technical, and can be found in [CV20, p 62-64]. The result of the Fenchel-Rockafeller theorem also provides us with a framework to develop a convergence criteria for numerical primal-dual methods. We will apply the duality gap as such a convergence criteria. The duality gap is the absolute value of the difference between the primal and dual solutions of primal-dual optimization problems. The duality gap is non-negative, and is zero if and only if we have a strong duality. Strong duality is exactly the second result of the Fenchel-Rockafeller theorem, that the optimal values of the primal and dual objective functions are identical. Thus, for the optimal solution to our dual problem (2.12) satisfying the requirements of Theorem 2.2, the duality gap will be zero.

The result from Theorem 2.2 will be used to establish optimality conditions for our problem, which are of the form presented in Equation (2.12). Together with the results from Theorem 2.1, we now have an approach as to how to derive the optimality conditions for the minimization problem.

2.6.3 Proximal point mappings

In order to perform numerical optimization and to do the necessary theoretical analysis to prepare for the numerical implementations of possible solution algorithms, we will need to define, understand and implement a type of mappings called proximal point mappings.

Definition 2.8. Let U be a Hilbert space, and let $f \in \Gamma_0(U)$. Then the **proximal point mapping** of f at u is $\operatorname{prox}_f : U \to U$ with

$$\operatorname{prox}_f(u) = \operatorname{argmin}_{\hat{u}\in U}\frac{1}{2}\|\hat{u}-u\|_2^2 + f(\hat{u}).$$

An interesting example of a proximal point mapping is that of the characteristic function onto a set C. In this case, the proximal point mapping is identical to the projection operator onto the set.

$$\operatorname{prox}_{i_C}(u) = \operatorname{argmin}_{\hat{u}\in U}\left[\frac{1}{2}\|\hat{u}-u\|_2^2 + i_c(\hat{u})\right] = \operatorname{argmin}_{\hat{u}\in C}\frac{1}{2}\|\hat{u}-u\|_2^2 = \pi_C(u)$$

Optimality conditions and their relation to proximal point operators

We still assume that $f \in \Gamma_0(U)$, and that $\operatorname{prox}_f : U \to U$. Then we can show the following about the proximal point operator:

$$\operatorname{prox}_f(u) = (I+\partial f)^{-1}(\hat{u}) \Longleftrightarrow u \in \hat{u} + \partial f(\hat{u}). \tag{2.13}$$

This is shown by exploiting that $\partial(f+h) = \partial f + \partial h$ for $h(\hat{u}) = \frac{1}{2}\|\hat{u}-u\|_2^2$. We know that $\hat{u}$ minimizes $(f+h)$ if and only if $0 \in \partial(f+h)(\hat{u}) = \partial f(\hat{u}) + \partial h(\hat{u})$. h is convex and differentiable, thus the only element in the sub-gradient $\partial h(\hat{u})$ is the gradient of h at $\hat{u}$, which is $\hat{u} - u$. Thus we obtain $0 \in \partial f(\hat{u}) + \hat{u} - u$, which is equivalent to

$$u \in \hat{u} + \partial f(\hat{u}) = (I + \partial f)\hat{u}.$$

And, directly from the definition of the proximal operator, we have proven (2.13).

Now let u_1, u_2 be elements of U, and let $\varepsilon > 0$ be an arbitrary positive real number. Then $u_2 \in \partial f(u_1)$ if and only if $u_1 = \operatorname{prox}_{\varepsilon f}(u_1 + \varepsilon u_2)$, as a direct consequence of Equation (2.13). We have thus arrived at a fixed point formulation yielding that u_1 is a fixed point of $\operatorname{prox}_{\varepsilon f}$ if and only if $0 \in \partial f(u_1)$. From Theorem 2.1 we recall that the latter occurs if and only if u_1 is a minimzer for f. This means that the fixed point will co-incide with the minimizer. This means that two optimality conditions for the minimizers $\hat{u} \in U$ have been derived:

$$0 \in \partial f(\hat{u}) \tag{2.14}$$
$$\hat{u} = \operatorname{prox}_{\varepsilon f}(\hat{u}) \tag{2.15}$$

2.6.4 The Chambolle-Pock algorithm

When solving primal-dual optimization problems numerically, it is necessary to use efficient numerical solvers. The method we will apply, is the *Chambolle-Pock algorithm* is a primal-dual algorithm, in which fixed point iterations are performed both for the primal variable u, and the dual variable p. We will refer to known convergence results for the algorithm, and parameter choices guaranteeing convergence will be explained and justified.

The method is based on a reasonably simple fixed point iteration for p and u, originally described by the Arrow-Hurwicz method. We let τ and σ be the stepsizes for each iteration for u and p respectively. The choices of these stepsizes influence the convergence of the method, and will be adressed in the paragraph concerning convergence. For a general minimization problem on the form $\min_u f(u) + g(Bu)$ where B is a linear operator, we can start by writing the fixed point iteration in the following manner:

$$\begin{aligned} p_{k+1} &\longleftarrow \operatorname{prox}_{\sigma g^*}(p_k + \sigma B u_k) \\ u_{k+1} &\longleftarrow \operatorname{prox}_{\tau f}(u_k - \tau B^* p_{k+1}) \end{aligned}$$

This method is known as the Arrow-Hurwicz method, and origins from work published by Kenneth Arrow and Leonid Hurwicz in 1956 [AH56]. More recent work, performed by Chambolle and Pock, tries to improve the Arrow-Hurwicz algorithm. It is this improved algorithm that will be used as the primary numerical solver throughout this text. The improvement is mainly done by introducing an intermediate variable $\hat{u}$, which extrapolates the current and previous iterates for u [CP11]. The Chambolle-Pock method then takes the form

$$\begin{aligned} p_{k+1} &\longleftarrow \operatorname{prox}_{\sigma g^*}(p_k + \sigma B \hat{u}_k) \\ u_{k+1} &\longleftarrow \operatorname{prox}_{\tau f}(u_k - \tau B^* p_{k+1}) \\ \hat{u}_{k+1} &\longleftarrow u_{k+1} + \theta(u_{k+1} - u_k) \end{aligned} \tag{2.16}$$

It is worth noting that the parameter choice $\theta = 0$ restores the Arrow-Hurwicz method.

It has been shown that a sufficient criteria to obtain convergence of the method is to choose the parameters τ and σ such that $\tau\sigma \leq \frac{1}{\|B\|^2}$ [CP11]. Thus, it is possible to simply choose $\tau = \sigma = \frac{1}{\|B\|}$, to ensure convergence of the Chambolle-Pock algorithm. These choices of τ and σ are not necessarily optimal, but they are never the less choices that ensures convergence.

3 Total variation denoising

A common regularization technique used to make images more smooth is total variation (TV). Total variation does only contain one regularization term, and thus also only one regularization parameter $\lambda \geq 0$. Total variation in one channel for a two-dimensional image u over the area Ω is defined in the following way.

$$\mathrm{TV}(u) = \int_{\Omega} \|\nabla u\| \mathrm{d}\mathbf{x} = \int_{\Omega} \sqrt{\frac{\partial u}{\partial x}^2 + \frac{\partial u}{\partial y}^2} \mathrm{d}\mathbf{x}. \tag{3.1}$$

Image denoising is however a discrete problem, so we will need to apply the discretized gradient operator $D : [0,1]^{mn} \to \mathbb{R}^{2mn}$ defined in Equation (2.9). Then, the formal definition of total variation for discrete images is given as

$$\mathrm{TV}(u) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sqrt{\Big(D_x(i,j)\Big)^2 + \Big(D_y(i,j)\Big)^2} = \|Du\|_1. \tag{3.2}$$

This is the same as the sum of all the mn pointwise Euclidean norms of the gradient Du of u, that is obtained by applying the operator D as defined in Equation (2.9) on u.

Total variation was introduced for noise removal problems in discrete images by Rudin, Osher and Fatemi in 1992 [ROF92]. Their denoising model is known as the Rudin-Osher-Fatemi-model (ROF) and uses discretized total variation with one parameter $\lambda \geq 0$. The minimization problem T_λ given the regularization parameter λ and the noisy measurement v can thus be expressed as

$$T_\lambda(v) = \min_{Au \in U} \Big[\frac{1}{2}\|u - v\|^2 + \lambda \mathrm{TV}(u)\Big]. \tag{3.3}$$

The ROF-model is considered an effective technique for image denoising, with advantages such as in particular smoothing of edges. The total variation approach has proven to be effective in combining this smoothing with noise removal in areas with fewer or less sharp edges [SC03].

We can note that $\lambda = 0$ means that no regularization is performed. In this case, the ROF-model reduces to a least squares minimization problem. Higher values of λ penalize edges in the image more, aiming to minimize the total variation defined in Equation (3.2).

One weakness that the total variation model suffers from, is that it may lead to a *staircasing* effect [BKP10]. Staircasing means that the reconstructed image is piecewise constant between the edges. The penalization of edges in the ROF-model can be 'blamed' for this feature. In general, total variation is well suited for regularization of images with few, but sharp edges [Hub+19]. If, however, the image contains more details and smaller edges, the staircasing effect is more likely to cause problems while regularizing.

For colour images with three RGB-channels, there are several approaches one can choose from when selecting the parameter λ. One may choose the same parameter in all channels, or one may treat the channels separately. The approach that will be followed throughout this master thesis is to treat the three RGB-channels separately in all numerical experiments. That is, the parameter selection in the three channels will be independent of each other, but follow the same algorithm and rules. In other words, in all numerical experiments we are performing the same process three times, one for each channel.

3.1 Convex conjugates

We recall the minimization problem for total variation denoising from Equation (3.3). If we write the total variation as the one-norm of the gradient Du instead, we obtain

$$T_\lambda(v) = \min_{u \in U} \left[\frac{1}{2} \|u - v\|^2 + \lambda \|Du\|_1 \right]. \tag{3.4}$$

We now want to split the right hand side into two terms. Therefore we introduce and define $f(u) = \|Au - v\|_2^2$ as the data discrepancy term, and $g(Du) = \lambda \|Du\|_1$ as the regularization term.

We want to use the notion of duality to solve the optimization problem, and to apply the Fenchel-Rockafeller theorem. In order to do that, we need to calculate the convex conjugates of the different terms. We start with f, assuming that the linear operator A is a non-singular and real-valued matrix. Note that when $A = I$, this condition is satisfied. First, we find an expression for the conjugate $f^*(p)$ directly from the definition of a convex conjugate that was given in Definition 2.4:

$$f^*(p) = \sup_{u \in U} \left[\langle u, p \rangle - \frac{1}{2} \|Au - v\|_2^2 \right]$$

The substitution $w = Au$ simplifies the problem to

$$f^*(p) = \sup_{\substack{w \in V \\ w = Au}} \left[\langle w, A^{-*} p \rangle - \frac{1}{2} \|w - v\|_2^2 \right]$$

Then, by differentiating this equation with respect to w to find its supremum and corresponding value $\hat{w}$, the first order optimality condition yields $A^{-*}p = \hat{w} - v$ and therefore $\hat{w} = A^{-*}p + v$. Substituting this, the expression for the convex conjugate $f^*(p)$ can now be written as:

$$\begin{aligned} f^*(p) &= \langle A^{-1}(A^{-*}p + v), p \rangle - \frac{1}{2} \|A^{-*}p + v - v\|_2^2 \\ &= \langle A^{-*}p + v, A^{-*}p \rangle - \frac{1}{2} \|A^{-*}p\|_2^2 \\ &= \frac{1}{2} \|A^{-*}p\|_2^2 + \langle v, A^{-*}p \rangle + \frac{1}{2} \|v\|_2^2 - \frac{1}{2} \|v\|_2^2 \\ &= \frac{1}{2} \|A^{-*}p + v\|_2^2 - \frac{1}{2} \|v\|_2^2, \end{aligned}$$

and thus we conclude that the convex conjugate of f is given as

$$f^*(p) = \frac{1}{2} \|A^{-*}p + v\|_2^2 - \frac{1}{2} \|v\|_2^2. \tag{3.5}$$

In the special case of denoising, where $A = I$, we can continue the calculation to obtain

$$f^*(p) = \frac{1}{2} \Big(\|p + v\|_2^2 - \|v\|_2^2 \Big).$$

In our total variation model, we also include the regularization term $g(Du) = \lambda \|Du\|_1$. Thus we also need to calculate the conjugate of $g(w) = \lambda \|w\|_1$. In order to simplify calculations, we introduce this useful lemma.

Lemma 3.1. *Let U be a Hilbert space and λ a non-zero real and positive number. Let $g : U \to \mathbb{R}$, and let $h = \lambda g$. Then $h^*(p) = \lambda g^*(\frac{p}{\lambda})$.*

Proof. For any $p \in U$, by definition

$$h^*(p) = \sup_{v \in U} \Big(\langle p, v \rangle - h(v) \Big).$$

From the definition of h, this is equivalent to

$$h^*(p) = \sup_{v \in U} \Big(\langle p, v \rangle - \lambda g(v) \Big).$$

Now we can use that inner products are linear in the first term and that λ is strictly positive to write

$$h^*(p) = \lambda \sup_{v \in U} \Big(\langle \frac{p}{\lambda}, v \rangle - g(v) \Big).$$

Now this supremum is exactly the convex conjugate of g evaluated in $\frac{p}{\lambda}$, and thus we conclude

$$h^*(p) = \lambda g^*(\frac{p}{\lambda}). \tag{3.6}$$

□

From the lemma above we know that to determine $g^*(p)$, we only need to determine the convex conjugate of $h(u) = \|u\|_1$, and then scale the argument according to the result from the lemma.

Again starting from Definition 2.4, we immediately obtain

$$h^*(p) = \sup_{u \in U} \Big[\langle u, p \rangle - \|u\|_1 \Big].$$

We can discretize this in each of the nm pixels:

$$h^*(p) = \sup_{u \in U} \Big[\sum_{i=1}^{n} \sum_{j=1}^{m} \langle u_{(i,j)}, p_{(i,j)} \rangle - \|u_{(i,j)}\|_2 \Big]. \tag{3.7}$$

From here we can apply Cauchy-Schwarz to obtain

$$h^*(p) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sup_{u_{(i,j)}} \|u_{(i,j)}\|_2 (\|p_{(i,j)}\|_2 - 1). \tag{3.8}$$

From Equation (3.8) we draw the following conclusion about the convex conjugate of h: If $\|p_{(i,j)}\|_2 \leq 1$ for all pixels $1 \leq i \leq n, 1 \leq j \leq m$, then the supremum of the right hand side is obtained by letting u be identically equal to zero. If, however, there exists at least one pixel for which $\|p_{(i,j)}\|_2 > 1$, the parenthesis is positive, and we can choose an u with an arbitrarily large norm to obtain a larger supremum. Thus

$$h^*(p) = \begin{cases} 0 & \text{if } \|p_{(i,j)}\|_2 \leq 1 \text{ for all } nm \text{ pixels} \\ \infty & \text{else.} \end{cases} \tag{3.9}$$

And by the lemma, we can scale this result for $h^*(p)$ to arrive at the following expression for $g^*(p)$:

$$g^*(p) = \begin{cases} 0 & \text{if } \|p_{(i,j)}\|_2 \leq \lambda \text{ for all } nm \text{ pixels} \\ \infty & \text{else.} \end{cases}$$

3.2 Chambolle-Pock algorithm for total variation

The aim of this chapter is to implement and compare some known numerical algorithms for solving the image denoising problem (2.2). The minimization problem under consideration is the one defined in (3.1). The derivation of the algorithm does to a great extent rely on the work that already have been done earlier in this text. The previously derived optimality conditions in Equations (2.14) and (2.15) give the fixed point iteration for p and u, with stepsizes σ and τ. The Chambolle-Pock method then takes the form

$$\begin{aligned} p_{k+1} &\longleftarrow \operatorname{prox}_{\sigma g^*}(p_k + \sigma D\hat{u}_k) \\ u_{k+1} &\longleftarrow \operatorname{prox}_{\tau f}(u_k - \tau D^* p_{k+1}) \\ \hat{u}_{k+1} &\longleftarrow u_{k+1} + \theta(u_{k+1} - u_k) \end{aligned} \tag{3.11}$$

It is this Chambolle-Pock method that will be used to solve the total variation minimization problem (3.1) for denoising, with regularization parameter λ. Choosing the best λ is key to obtain good reconstructions of the original image. Different parameter choice techniques will be used, in order to identify the optimal parameter for the total variation denoising problem.

3.2.1 Calculation of prox-solutions

In order to implement the Chambolle-Pock algorithm described in Equation (3.11) properly, it is necessary to determine how to find the exact proximal point mappings that are used in the algorithm.

Prox operator τf

$$\operatorname{prox}_{\tau f}(u - \tau D^* p) = \operatorname*{arg\,min}_{w \in U} \Big[\frac{1}{2}\|w - u + \tau D^* p\|_2^2 + \frac{\tau}{2}\|Aw - v\|_2^2\Big]. \tag{3.12}$$

This is a simple minimization problem, and we can continue by differentiating the functional with respect to w without major difficulties. The gradient is given as

$$w - u + \tau D^* p + \tau A^*(Aw - v).$$

We then use the first order necessary optimality condition for the simple mimimization problem in Equation (3.12), and set the derivative equal to zero. We then solve the linear equation, and obtain the following:

$$(I + \tau A^* A)w = u - \tau D^* p + \tau A^* v.$$

In the case of denoising, with $A = I = A^*$, this can be simplified to

$$\operatorname{prox}_{\tau f}(u - \tau D^* p) = \frac{u - \tau D^* p + \tau v}{1 + \tau}. \tag{3.13}$$

In the numerical experiments, we will consider denoising, and thus use the prox operator found in Equation (3.13). In a more general case, where the linear operator $A : U \to V$ is different from the identity operator, we obtain the following expression for the prox operator:

$$\text{prox}_{\tau f}(u - \tau D^* p) = (I + \tau A^* A)^{-1}(u - \tau D^* p + \tau A^* v). \tag{3.14}$$

This requires $(I + \tau A^* A)$ to be invertible, which we can show in the following manner: $A^* A$ is symmetric, and so is the identity matrix. Thus $(I + \tau A^* A)$ is also symmetric. We know that all positive definite symmetric matrices have only positive and real eigenvalues, and therefore are invertible. So we only need to show that $(I + \tau A^* A)$ is positive definite. This is done by noting that for any $u \neq 0$,

$$u^T (I + \tau A^* A)u = u^T Iu + \tau u^T A^* Au = u^T u + \tau u^T A^* Au = \|u\|^2 + \tau \|Au\|^2 > 0.$$

Prox operator σg^*

If we now turn our attention to the proximal point mapping of σg^*, we obtain the following expression from the definition of these mappings:

$$\text{prox}_{\sigma g^*}(p_k - \sigma D u_k) = \arg\min_q \frac{1}{2}\|q - p_k - \sigma D u_k\|_2^2 + \sigma g^*(q). \tag{3.15}$$

To simplify the notation, we remove the index k which denotes the iteration number, as the results are valid regardless of how many previous iterations that have been performed. We thus have this expression for the proximal point mapping of σg:

$$\text{prox}_{\sigma g^*}(p - \sigma D u) = \arg\min_q \frac{1}{2}\|q - p - \sigma D u\|_2^2 + \sigma g^*(q). \tag{3.16}$$

This can be calculated pointwise in each pixel (i, j) for $1 \leq i \leq n$ and $1 \leq j \leq m$. Moreover, for the sake of simplicity, we introduce w, which is defined as

$$w_{(i,j)} = p_{(i,j)} + \sigma D(u_{(i,j)}).$$

The minimization problem (3.16) can now be rewritten pointwise as

$$\arg\min_{q_{(i,j)}} \Big[\frac{1}{2}\|q_{(i,j)} - w_{(i,j)}\|_2^2 + \sigma g^*(q_{(i,j)})\Big], \tag{3.17}$$

again being valid for all the nm pixels in the image. We recall that the value of $g^*(q_{(i,j)})$ depends on the infinity norm of the different Euclidean norms. If $|q_{(i,j)}|_2 \leq \lambda$ in all pixels, then $g^*(q_{(i,j)}) = 0$. Otherwise, $g^*(q_{(i,j)}) = +\infty$. This means that the pointwise minimization problem now can be rewritten again, this time as

$$\arg\min_{\|q_{(i,j)}\|_2 \leq \lambda} \|q_{(i,j)} - w_{(i,j)}\|_2^2. \tag{3.18}$$

If the Euclidean norm of $w_{(i,j)}$ is smaller than or equal to λ, then $w_{(i,j)}$ itself solves the pointwise minimization problem (3.18), and $q_{(i,j)} = w_{(i,j)} = p_{(i,j)} + \sigma D(u_{(i,j)})$. In the case where the Euclidean norm of $w_{(i,j)}$ is bigger than λ, we need to scale the vector, so that it is contained in the convex set B_λ. Intuitively, we think of keeping the direction of the vector, but scaling the length of the vector down to λ. In Lemma 4.1, we will prove that this intuition is correct for convex balls. For now however, we will simply state the result, which gives us that $q_{(i,j)} = \lambda \frac{w_{(i,j)}}{\|w_{(i,j)}\|_2}$.

So, to conclude the calculation, we can summarize the pointwise prox operator on g^* as

$$\text{prox}_{\sigma g^*}(p - \sigma Du)_{(i,j)} = \begin{cases} p_{(i,j)} - \sigma Du_{(i,j)}, & \text{for } \|p_{(i,j)} - \sigma Du_{(i,j)}\|_2 \leq \lambda \\ \lambda \frac{p_{(i,j)} - \sigma Du_{(i,j)}}{\|p_{(i,j)} - \sigma Du_{(i,j)}\|_2}, & \text{for } \|p_{(i,j)} - \sigma Du_{(i,j)}\|_2 > \lambda \end{cases}, \tag{3.19}$$

and thus the prox operator for both f and g^* has been determined, and are given in Equations (3.13) and (3.19) respectively. These are then used as proximal point operators in the Chambolle-Pock algorithm, see Equation (3.11).

3.2.2 The complete Chambolle-Pock algorithm for TV

We have now calculated the proximal operators that are necessary to perform all steps in the Chambolle-Pock algorithm that was described in Equation (3.11). We are thus ready to describe the procedure that will be used to denoise images with total variation as the only regularization term. The procedure requires the regularization parameter λ to be chosen. The noisy signal v is also inputted to the numerical algorithm, which takes the following form:

```
Choose $\theta$ for Chambolle-Pock method
Choose steplengths $\tau$ and $\sigma$
$u, u_{\text{prev}}, \hat{u} \leftarrow v$
$p \leftarrow \mathbf{0}$
while not converged do
    $u_{\text{prev}} \leftarrow u$
    $\hat{p} \leftarrow p + \sigma D(\hat{u})$
    $p \leftarrow \text{prox}_{\sigma g^*}(\hat{p})$
    $\nabla p \leftarrow D^*(p)$
    $u \leftarrow \text{prox}_{\tau f}(u + \tau \nabla p + \tau v)$
    $\hat{u} = u + \theta(u - u_{\text{prev})}$
end while
return u
```

Algorithm 1: TV denoising algorithm

We see that in each iteration we start by updating the dual variable p, and then scale it onto to convex set B_λ. We then use the update of the dual variable to update the primal variable u, and again we apply the proximal point mapping on the suggested update. The algorithm runs until the pre-selected convergence criteria is reached, or until it is stopped for other reasons. In the next subsections we will discuss the convergence of Chambolle-Pock for total variation, and also develop a criteria for determining whether the algorithm has converged or not.

3.2.3 Convergence of Chambolle-Pock algorithm for TV

Recall that we need to have $\tau\sigma \leq \frac{1}{\|D\|^2}$ in order to guarantee convergence of the Chambolle-Pock algorithm for the primal problem $\min_u f(u) + g(Du)$. So we need to calculate or estimate the operator norm of D, the gradient operator we apply in the total variation term. From the definition on operator norms we have that $\|D\| = \sup\{\|Du\| : u \in \mathbb{R}^{nm} \text{ and } \|u\| = 1\}$. Here the operator D is applied on the three different RGB-channels separately. This explains that we operate in $\mathbb{R}^{nm}$ and not in $\mathbb{R}^{3nm}$. Moreover, we can see that we always have that $\|D\| < \sqrt{8}$ [Lud20]. Thus, the convergence criteria can be written as $\tau\sigma \leq \frac{1}{8}$, and choices of τ and σ must be made in accordance with this criteria.

As convergence criteria for the numerical implementation, the relative duality gap, compared to the initial duality gap, is used with tolerance 10^{-6}. The duality gap is calculated as

$$\frac{1}{2}\|u - v\|^2 + \lambda TV(u) - \frac{\|v\|^2}{2} + \frac{1}{2}\|f - D^* p\|^2, \tag{3.20}$$

where the two first terms come from the regularization functional, as the solution of the primal problem. Similarly, the two other terms come from the dual problem, and is the solution of this. The duality gap is then the difference between the solution to the primal and the dual problem, and is used as a convergence criteria in accordance with the Fenchel-Rockafeller theorem (Theorem 2.2).

Also, for bad parameter choices, we simply end the algorithm after 100 iterations if the convergence criteria is not yet met. This saves runtime and also in-depth calculations of many non-optimal parameter choices.

4 Total generalized variation (TGV) denoising

Total generalized variation (TGV) is a more recent regularization functional used for denoising of images and other digital signals. Unlike the total variation method, TGV is a denoising algorithm of second order. That is, the TGV functional that will be considered in this text contains derivatives of second order [Li+20]. A general TGV method of order k contains approximations of derivatives up to the order k, and was introduced by Bredies, Kunisch and Pock in 2009 [BKP10].

Total generalized variation has increasingly popular image denoising applications within medicine, for example in magnetic resonance [Kno+10] and tomography [Hub+19]. These are both technologies for which removing noise, blur or systemic measuring errors may be essential to improving the possibility to, manually or by a machine learning algorithm, discover illnesses or medical abnormalities [LL19].

A major advantage with TGV over TV is that the staircasing effect is reduced, compared to the ROF-model in total variation denoising. [BKP10]. However, the TGV regularization functional is more complicated. This affects both the runtime and the efficiency of numerical algorithms. Moreover, TGV denoising also requires two parameter choices. In order to optimize the performance of TGV denoising, it is thus necessary to perform multi-parameter optimization, which is more difficult than single-parameter optimization.

Recalling from Equation (2.5) that after introducing Tikhonov regularization terms, we can in general express the optimization problem as a minimization problem containing a data discrepancy term plus some regularization terms. In the case of total generalized variation, we introduce a regularization functional $TGV(u)_2$. The index two indicates that the total generalized variation is of order two. The regularization functional to be considered does now take two variables, u and $\mathbf{w}$. $u(x, y)$ is, as before, the signal value of the image in each of the mn pixels. The vector-valued function $\mathbf{w}(x, y) : \mathbb{R}^2 \to \mathbb{R}^2$ can be considered as an approximation to the gradient $\nabla u(x, y)$ and is denoted by $\mathbf{w}(x, y) = (w_1(x, y), w_2(x, y))$. We write the regularization functional as a functional with two variables in the following way.

$$R(u, \mathbf{w}) = \lambda_1 \int_\Omega |\nabla u - \mathbf{w}| \mathrm{d}x + \lambda_2 \int_\Omega |\mathcal{E}\mathbf{w}| \mathrm{d}x, \tag{4.1}$$

where the symmetrised gradient of the vector-valued function $\mathbf{w}$ is expressed as in [GN20]:

$$\mathcal{E}\mathbf{w} = \frac{1}{2}(\nabla \mathbf{w} + (\nabla \mathbf{w})^T). \tag{4.2}$$

We are going to apply total generalized variation on finite-dimensional and discrete signals when we perform image denoising, so we will need a discretized version of the regularization terms $R(u, \mathbf{w})$. We will denote the discretized version of the regularization terms by $\mathrm{TGV}(u, \mathbf{w})$, and it will take the following form:

$$\mathrm{TGV}(u, \mathbf{w}) = \lambda_1 \sum_{i=1}^{n} \sum_{j=1}^{m} |\nabla u_{i,j} - \mathbf{w}_{i,j}| + \lambda_2 \sum_{i=1}^{n} \sum_{j=1}^{m} |\mathcal{E}\mathbf{w}_{i,j}|. \tag{4.3}$$

The gradient $\nabla \mathbf{w}$ is given as

$$\nabla \mathbf{w} = \begin{bmatrix} \frac{\partial w_1}{\partial x} & \frac{\partial w_1}{\partial y} \\ \frac{\partial w_2}{\partial x} & \frac{\partial w_2}{\partial y} \end{bmatrix},$$

and in the discrete case we want to express this gradient by the forward difference operators D_x and D_y.

$$\nabla \mathbf{w} = \begin{bmatrix} D_x(w_1) & D_y(w_1) \\ D_x(w_2) & D_y(w_2) \end{bmatrix}, \tag{4.4}$$

From Equation (4.4) and the definition of the symmetrized gradient in Equation (4.2), we can express $\mathcal{E}\mathbf{w}$ as

$$\mathcal{E}\mathbf{w} = \frac{1}{2} \begin{bmatrix} 2D_x(w_1) & D_y(w_1) + D_x(w_2) \\ D_y(w_1) + D_x(w_2) & 2D_y(w_2) \end{bmatrix}. \tag{4.5}$$

In each of the nm pixels (i, j), $\mathcal{E}\mathbf{w}_{(i,j)}$ is a symmetric matrix in $\mathbb{R}^{2\times 2}$. Symmetric matrices contain no more than three unique values. Thus, the matrix can equivalently be stored as a vector in $\mathbb{R}^3$. This saves memory and runtime in numerical simulations, and does not affect the results of the analytical calculations.

Combining the regularization terms with the data discrepancy term, we obtain the following minimization problem for second order total generalized variation:

$$\min_{u} \Big[L(u, v) + \mathrm{TGV}_2(u) \Big] = \min_{u,\mathbf{w}} \Big[L(u, v) + \mathrm{TGV}(u, \mathbf{w}) \Big].$$

We denote the last minimization problem by $T_{\boldsymbol{\lambda}}(v)$ for a fixed vector $\boldsymbol{\lambda} \in \mathbb{R}^2_{(+)}$ of regularization parameters. The complete minimization problem is then expressed as

$$T_{\boldsymbol{\lambda}}(v) = \min_{u,\mathbf{w}} \Big[L(u, v) + \mathrm{TGV}(u, \mathbf{w}) \Big], \tag{4.6}$$

where v is the noisy data. And explicitly, the discrete minimization problem can be written as

$$\hat{u}, \hat{\mathbf{w}} = \arg\min_{u,\mathbf{w}} \Big(\frac{1}{2} \|u - v\|^2 + \lambda_1 \sum_{i=1}^{n} \sum_{j=1}^{m} \|\nabla u_{(i,j)} - \mathbf{w}_{(i,j)}\|_2 + \lambda_2 \sum_{i=1}^{n} \sum_{j=1}^{m} \|\mathcal{E}\mathbf{w}_{(i,j)}\|_2. \Big) \tag{4.7}$$

Note that this is a multi-parameter parameter selection problem, for which both λ_1 and λ_2 have to be chosen. This is more complicated than a single-parameter parameter selection problem.

The functional in Equation (4.7) includes three terms. The first term is the data discrepancy term, measuring the mean square error between the reconstructed and noisy signal.

The second term contains the norm of the difference between the discretized forward gradient ∇u of the signal and an approximation $\mathbf{w}$ to this gradient. Where we in total variation try to minimize the norm of the gradient itself, this new approach instead aims to minimize $\|\nabla u - \mathbf{w}\|_2$. Consequently, the reconstructed signal will not be piecewise constant, but may to some extent be piecewise linear.

The third term is the norm of the socalled **symmetrized gradient** of $\mathbf{w}$, which is defined above. Why would one want to minimize the norm of the symmetrized gradient of an approximation of the gradient of the reconstructed signal? This symmetrized gradient can be considered as an approximation of the second derivative of u. By including this term as a separate regularization term, we also require the (approximation to) the second derivative to be regular. Moreover, the fraction $\frac{\lambda_1}{\lambda_2}$ determines how the first and second derivatives are balanced in the regularized functional[Kno+10]. It is not intuitively clear how this works and what the effects are. In Appendix B, an example for a one-dimensional signal is included. This example can, if needed, be used to get a stronger intuition of how the regularization terms in the TGV denoising functional work together.

The parameters λ_1 and λ_2 respectively are used for weighing the second and third term. Optimal choices of these parameters will vary from signal to signal, and it is not trivial to determine which parameters are optimal. We can again note that the choice $\lambda_1 = \lambda_2 = 0$ will reduce the problem (4.6) to a least squares problem. The main aim of this text is to develop automated parameter choice methods for these choices, and to evaluate the performance of the different methods. We will moreover compare denoising results with TGV multi-parameter regularization with those from TV single-parameter denoising.

4.1 Convex conjugates

We need to calculate the convex conjugates of the different terms in the total generalized variation functional, in order to develop a numerical primal-dual solver similar to the one for total variation. We note that the TGV-functional takes two primal variables u and $\mathbf{w}$, and that we thus also need two dual variables.

Again, we start by splitting the terms of $T_{\boldsymbol{\lambda}}(u, \mathbf{w})$ into different functionals $F(u, \mathbf{w})$ and $G(A_1(u, \mathbf{w}), A_2(u, \mathbf{w}))$, where A_1 and A_2 are the linear operators $\nabla u - \mathbf{w}$ and $\mathcal{E}\mathbf{w}$. Thus we have that $G(y, z) = \lambda_1 \|y\|_1 + \lambda_2 \|z\|_1$, and can express the primal functional in the form $F(u, \mathbf{w}) + G(A_1(u, \mathbf{w}), A_2(u, \mathbf{w}))$. We denote the data discrepancy term by $F(u, \mathbf{w}) = \|u - v\|_2^2$.

The data discrepancy term does not depend on $\mathbf{w}$, a fact that can be exploited when calculating the convex conjugate. We can express $F(u, \mathbf{w}) := F_1(u) + F_2(\mathbf{w}) = \|u - v\|_2^2 + 0$. Here F_1 is the mean squared difference between the reconstructed and the noisy signal, while F_2 is identically equal to zero. We can use this 'split' to determine the convex conjugate of F by considering F_1 and F_2 as two separate functions of one variable each, depending on u and $\mathbf{w}$ respectively.

$$F^*(p, q) = \sup_{u \in U} \Big(\langle u, p \rangle - F_1(u) \Big) + \sup_{\mathbf{w} \in \mathbb{R}^{2mn}} \Big(\langle \mathbf{w}, q \rangle - F_2(\mathbf{w}) \Big).$$

The first supremum is identical to the one we obtained when performing calculations for total variation, and has the same conjugate. Thus we obtain, in the case of pure denoising,

$$F^*(p, q) = \frac{1}{2}\|p\|_2^2 + \langle p, v \rangle + \sup_{\mathbf{w}\mathbb{R}^{2mn}} \Big(\langle \mathbf{w}, q \rangle - 0 \Big).$$

The second supremum will be infinite for any $q \neq 0$. If $q = 0$, then naturally also $\langle \mathbf{w}, q \rangle - 0 = 0$. This leaves us with the following expression for $F^*(p, q)$:

$$F^*(p, q) = \begin{cases} \frac{1}{2}\|p\|_2^2 + \langle p, v \rangle & \text{if } q = 0 \\ \infty & \text{if } q \neq 0. \end{cases} \tag{4.8}$$

The convex conjugates of the two regularization terms must also be determined. We notice that the second and the third term in the TGV-functional (4.7) both consist of a (weigthed) one-norm over Ω. The results from the regularization term in total variation can thus be applied also on these two terms, when calculating the convex conjugates. The result we have is that for a general $G(\psi) = \lambda\|\psi\|_1$ with $\lambda \geq 0$, the convex conjugate G^* is given as $G^*(s) = \mathcal{I}_{\|\cdot\|_\infty \leq \lambda}(s)$[Kno+10].

From Equation (4.1) we observe that the regularization functional is a combination of two terms of this form. We can consider the regularization functional R as a sum of two regularization terms R_1 and R_2 depending on different and independent variables. Consequently, the convex conjugate of R is also given as the sum of the convex conjugates of R_1 and R_2. This gives us $G^*(s,t) = \mathcal{I}_{\|\cdot\|_\infty \leq \lambda_1}(s) + \mathcal{I}_{\|\cdot\|_\infty \leq \lambda_2}(t)$.

This means that we have the following four primal and dual functionals.

$$\begin{aligned}
F(u,\mathbf{w}) &= \|u-v\|_2^2 \\
G(y,z) &= \lambda_1\|y\|_1 + \lambda_2\|z\|_1 \\
F^*(p,q) &= \begin{cases} \frac{1}{2}\|p\|_2^2 + \langle p, v\rangle & \text{if } q = 0 \\ \infty & \text{if } q \neq 0. \end{cases} \\
G^*(s,t) &= \mathcal{I}_{\|\cdot\|_\infty \leq \lambda_1}(s) + \mathcal{I}_{\|\cdot\|_\infty \leq \lambda_2}(t)
\end{aligned}$$

For the dual functional G^* with dual variables p and q, we have that the dual functional is the sum of two indicator functions on balls centered at the origin with radiuses of λ_1 and λ_2 respectively. In the numerical algorithm, each update of these variables will require that the update is projected onto these convex sets.

Projections onto the convex sets

The dual variables p and q are both required to be elements of the convex sets $P = B_{\lambda_1}$ and $Q = B_{\lambda_2}$. Both the convex sets are closed balls centered at the origin, with radiuses of λ_1 and λ_2 respectively. This means that updated iterations of p and q need to be projected onto the convex sets. Projections of vectors onto balls centered at the origin is among the easiest projections to calculate. In order to make the result valid generally, we denote by B_R a closed ball centered at the origin with radius $R > 0$, and introduce the following lemma.

Lemma 4.1. Let $x \in \mathbb{R}^n$ be an arbitrary vector. The projection of the vector onto the ball B_λ is given by

$$\operatorname{proj}_{B_\lambda}(x) = \frac{x}{\max(1, \frac{\|x\|}{\lambda})}. \tag{4.9}$$

Proof. To determine the projection, we are concerned with the minimization problem

$$\operatorname{proj}_{B_\lambda}(x) = \underset{\|q\|_2^2 \leq \lambda^2}{\arg\min}\|q-x\|_2^2. \tag{4.10}$$

We can start by consider the Lagrangian of the problem with the Lagrangian parameter γ.

$$\mathcal{L}(q,\gamma) = \|q-x\|_2^2 + \gamma(\|q\|_2 - \lambda). \tag{4.11}$$

The KKT-conditions for this problem are given as

$$\frac{\partial \mathcal{L}}{\partial q} = 2(q-x) + 2\gamma q = 0 \tag{4.12}$$

$$\gamma(\|q\|_2 - \lambda) = 0 \tag{4.13}$$

$$\gamma \geq 0 \tag{4.14}$$

$$\|q\|_2 - \lambda \leq 0. \tag{4.15}$$

In the case where $\gamma = 0$, we get from the first KKT-condition that $q = x$. This is the situation where $\|x\|_2 \leq \lambda$ and x itself is an element of the closed Euclidean ball B_λ. The projection of an element x on a set the element in fact is a member of, is of course the element x itself.

It is thus more interesting to consider the case when $\gamma > 0$. Then necessarily, from the second KKT-condition, $\|q\|_2 = \lambda$. That is, q is an element on the boundary of B_λ. If we again consider the first KKT-condition, and multiply from left with q^T, we obtain

$$(2 + 2\gamma)q^T q = 2q^T x.$$

We can simplify by dividing by two and noting that $q^T q = \|q\|_2^2 = \lambda^2$. Then

$$(1+\gamma)\lambda^2 = q^T x, \tag{4.16}$$

and

$$\gamma = \frac{q^T x - \lambda^2}{\lambda^2} > 0. \tag{4.17}$$

From the Cauchy-Schwarz inequality we have that $\|q\|_2 \|x\|_2 \geq q^T x$. When we combine this with the knowledge that $\|q\|_2 = \lambda$, we obtain

$$\frac{\lambda \|x\|_2 - \lambda^2}{\lambda^2} > 0,$$

and thus $\|x\|_2 > \lambda$. This means that $\gamma > 0$ if and only if $\|x\|_2 > \lambda$. Now we can use the expression for γ from Equation (4.17), by inserting it into the first KKT condition.

$$q + \frac{q^T x - \lambda^2}{\lambda^2} q = x$$

This can be written as

$$(1 + \frac{q^T x}{\lambda^2} - 1)q = x,$$

or even shorter as

$$\frac{q^T x}{\lambda^2} q = x. \tag{4.18}$$

Thus we have established that q and x are parallel to eachother in the case where $\|x\|_2 > \lambda$. We define the constant α such that $q = \alpha x$. Then we insert this into Equation (4.18) in order to determine α.

$$\frac{(\alpha x^T x)\alpha x}{\lambda^2} = x.$$

From here we observe that

$$\alpha^2 = \frac{\lambda^2}{\|x\|_2^2} < 1. \tag{4.19}$$

This means that in the case where $\|x\|_2 > \lambda$, the element of the ball with the shortest distance to x is given as $q = \frac{\lambda x}{\|x\|_2}$. The projection of x on B_λ can thus be written as

$$\text{proj}_{B_\lambda}(x) = \begin{cases} x, \text{ if } \|x\| \leq \lambda \\ \frac{\lambda x}{\|x\|}, \text{ if } \|x\| > \lambda, \end{cases}$$

or even more compact as

$$\text{proj}_{B_\lambda}(x) = \frac{x}{\max(1, \frac{\|x\|}{\lambda})}, \tag{4.20}$$

which is exactly what we wanted to prove, and thus the proof is complete. □

In each channel, we have that $p \in \mathbb{R}^{2nm}$ and that $q \in \mathbb{R}^{3nm}$. We can consider the pointwise vectors $p_{(i,j)} \in \mathbb{R}^2$ and $q_{(i,j)} \in \mathbb{R}^3$ for all the discrete pixels. We denote $p_{(i,j)} = \left[p^{(1)}_{(i,j)}, p^{(2)}_{(i,j)}\right]^T$. This is a vector for which the Euclidean norm is given in the 'normal' way, and we can express the pointwise norm of the vector as

$$\|p_{(i,j)}\|_2 = \sqrt{{p^{(1)}_{(i,j)}}^2 + {p^{(2)}_{(i,j)}}^2}. \tag{4.21}$$

For q, however, the norm is not that straight forward. In each pixel (i,j) we denote the elements of the three-dimensional vector by $q^{(1)}_{(i,j)}, q^{(2)}_{(i,j)}, q^{(3)}_{(i,j)}$. This vector is in the same space as the elements of the symmetric matrix $(\mathcal{E}w)_{(i,j)}$. This space has a somewhat unusual norm, to make sure that the change from a 2 × 2matrix to a three-dimensional vector does not change the value of the norm. This is however a property that is easily preserved, because the matrix is symmetrical. We require that $\|(\mathcal{E}w)_{(i,j)}\|_F = \|q_{(i,j)}\|_2$ in each pixel. Then

$$\|q_{(i,j)}\|_2 = \sqrt{{q^{(1)}_{(i,j)}}^2 + {q^{(2)}_{(i,j)}}^2 + 2{q^{(3)}_{(i,j)}}^2}. \tag{4.22}$$

With these norms established, and the result from Lemma (4.1), we arrive at the following pointwise projections on P and Q:

$$\text{proj}_P(p)_{(i,j)} = \text{proj}_{B_{\lambda_1}}(p_{(i,j)}) = \frac{p}{\max\left(1, \frac{\|p_{(i,j)}\|_2}{\lambda_1}\right)} \tag{4.23}$$

$$\text{proj}_Q(q)_{(i,j)} = \text{proj}_{B_{\lambda_2}}(q_{(i,j)}) = \frac{q}{\max\left(1, \frac{\|q_{(i,j)}\|_2}{\lambda_2}\right)} \tag{4.24}$$

4.2 Adjoint calculations

We now return to the linear operators A_1 and A_2, and need to calculate their adjoints. These adjoint linear operators are used in the dual functional, in accordance with the Fenchel-Rockafeller theorem. Recall that we denote by $A_1(u, \mathbf{w}) = \nabla u - \mathbf{w}$ the vector of which the one-norm is penalized in the first regularization term. The immediate observation is that $A_1 : \mathbb{R}^{mn} \times \mathbb{R}^{2mn} \to \mathbb{R}^{2mn}$. Thus we know that $A_1^* : \mathbb{R}^{2mn} \to \mathbb{R}^{mn} \times \mathbb{R}^{2mn}$. Moreover, from the definition given in Definition 2.4, we have that $\langle A_1(u, \mathbf{w}), p\rangle = \langle (u, \mathbf{w}), A_1^*(p)\rangle$. When calulating this adjoint operator, we split the operator A_1 into its two linear terms.

$$\langle A_1(u,\mathbf{w}),p\rangle = \langle \nabla u, p\rangle - \langle \mathbf{w}, p\rangle.$$

The adjoint operator of the discrete forward difference gradient operator ∇ is ∇^*, and we can now rewrite the expression as

$$\langle A_1(u,\mathbf{w}),p\rangle = \langle u, \nabla^* p\rangle - \langle \mathbf{w}, p\rangle.$$

Finally we combine these term and write out the resulting inner product.

$$\langle A_1(u,\mathbf{w}),p\rangle = \langle (u,\mathbf{w}), (\nabla^* p, -p)\rangle,$$

which implies that $A_1^*(p) = (\nabla^* p, -p)$.

For the second regularization term, we denote $A_2(u,\mathbf{w}) = \mathcal{E}\mathbf{w}$. We want to find the adjoint operator A_2^*, a process which also includes calculating the adjoint of the symmetrised gradient operator.

$$\begin{aligned}\langle A_2(u,\mathbf{w}),q\rangle =& \langle \mathcal{E}\mathbf{w}, q\rangle \\ =& \langle \mathbf{w}, \mathcal{E}^* q\rangle\end{aligned}$$

And thus $A_2^*(q) = (0, \mathcal{E}^* q)$. We will need to calculate the operator $\mathcal{E}^*$, and apply this on the dual variable $q \in \mathbb{R}^{3nm}$.

Calculation of $\mathcal{E}^*$

From the definition of an adjoint operator, we have that $\langle \mathcal{E}\mathbf{w}, q\rangle = \langle \mathbf{w}, \mathcal{E}^* q\rangle$. This property is essential when we determine how to implement the latter operator, $\mathcal{E}^*$. From Equation (4.5) we recall that

$$\mathcal{E}\mathbf{w} = \frac{1}{2}\begin{bmatrix} 2\frac{\partial w_1}{\partial x} & \frac{\partial w_1}{\partial y} + \frac{\partial w_2}{\partial x} \\ \frac{\partial w_1}{\partial y} + \frac{\partial w_2}{\partial x} & 2\frac{\partial w_2}{\partial y}\end{bmatrix}.$$

By inserting this expression directly and recalling that the symmetric matrix $\mathcal{E}\mathbf{w}$ can be treated as a vector in $\mathbb{R}^3$, we obtain an expression for the inner product $\langle \mathcal{E}\mathbf{w}, q\rangle$.

$$\langle \mathcal{E}\mathbf{w}, q\rangle = \langle \frac{\partial w_1}{\partial x}, q_1\rangle + \langle \frac{\partial w_2}{\partial y}, q_2\rangle + \langle \frac{1}{2}(\frac{\partial w_1}{\partial y} + \frac{\partial w_2}{\partial x}), 2q_3\rangle.$$

We have that $\mathcal{E} : \mathbb{R}^{2mn} \to \mathbb{R}^{3mn}$. Thus, we need to determine the adjoint operator $\mathcal{E}^* : \mathbb{R}^{3mn} \to \mathbb{R}^{2mn}$. We start by rewrting and discretizing the partial derivatives in the inner products by applying the operators D, D_x and D_y previously defined in Equations (2.7) to (2.9). Then the discretized inner product is rewritten as

$$\begin{aligned}\langle \mathcal{E}\mathbf{w}, q\rangle &= \langle D_x w_1, q_1\rangle + \langle D_y w_2, q_2\rangle + \langle D_y w_1, q_3\rangle + \langle D_x w_2, q_3\rangle \\ &= \langle w_1, D_x^* q_1\rangle + \langle w_2, D_y^* q_2\rangle + \langle w_1, D_y^* q_3\rangle + \langle w_2, D_x^* q_3\rangle.\end{aligned}$$

If we consider the terms containing w_1 and w_2 separately, this can be rewritten again as

$$\langle \mathcal{E}\mathbf{w}, q\rangle = \langle w_1, D_x^* q_1 + D_y^* q_3\rangle + \langle w_2, D_y^* q_2 + D_x^* q_3\rangle.$$

Finally, from the definition of an adjoint operator, this leads to

$$\langle \mathbf{w}, \mathcal{E}^* q \rangle = \langle w_1, D_x^* q_1 + D_y^* q_3 \rangle + \langle w_2, D_y^* q_2 + D_x^* q_3 \rangle$$

and

$$\mathcal{E}^* q = \Big(D_x^* q_1 + D_y^* q_3, D_y^* q_2 + D_x^* q_3 \Big). \tag{4.25}$$

Here D_x^* and D_y^* are the backward difference operators in the x- and y-direction respectively. Similar to the forward difference operators defined in equations (2.7) and (2.8), these operators both are operators from $\mathbb{R}^{nm}$ to $\mathbb{R}^{nm}$, and their definitions are given below.

$$D_x^*(u_{(i,j)}) = \begin{cases} u_{(i,j)} - u_{(i-1,j)} \text{ for } i > 0 \\ 0 \text{ for } i = 0 \end{cases} \tag{4.26}$$

$$D_y^*(u_{(i,j)}) = \begin{cases} u_{(i,j)} - u_{(i,j-1)} \text{ for } j > 0 \\ 0 \text{ for } i = 0 \end{cases} \tag{4.27}$$

4.3 Numerical TGV algorithm

With this framework established, we are ready to develop and present the numerical approach we will apply for total generalized variation. The primal-dual algorithm itself follows the same idea as we saw for total variation, and is also based on Chambolle-Pock. We will need to define a denoising algortihm based on the calculations of projections, proximal operators and convex conjugates from the last subsections, and also to determine the duality gap of the algorithm in order to develop a convergence criteria.

The numerical algorithm that has been implemented is the one suggested in [Kno+10], with a few minor modifications. Most importantly, we still treat each RGB-channel separately in the numerical algorithm. The input to the numerical algorithm is the noisy signal (or measurement) $v \in \mathbb{R}^{3nm}$, but split into three vectors of length nm, because of the channel-wise approach we have chosen. As for the total variation denoising, we also terminate the algorithm after R steps if it has not yet converged, to avoid using too much time on performing denoising for bad parameter choices.

4.3.1 Proximal point mappings for TGV

We can see from the numerical denoising algorithm that proximal point mappings are present also for total generalized variation, and that we need to calculate these mappings. We can reuse much of the work performed for total variation, but there are also some differences. The biggest difference from the calculations we performed in Section 3 is that we now have two primal variables, u and $\mathbf{w}$. Again, we will split the terms of the minimization problems, and consider each of the terms separately.

We first consider the data discrepancy term $F(u, \mathbf{w})$, and follow the definition of proximal point mappings given in Definition 2.8.

$$\text{prox}_{\tau F}(u, \mathbf{w}) = \underset{\hat{u}, \hat{\mathbf{w}}}{\arg\min} \Big(\frac{1}{2}(\|\hat{u} - u\|_2^2 + \|\hat{\mathbf{w}} - \mathbf{w}\|_2^2) + \tau F(\hat{u}, \hat{\mathbf{w}}) \Big). \tag{4.28}$$

```
Choose steplengths $\tau$ and $\sigma$
$u, \bar{u} \leftarrow v$
$\mathbf{w}, \bar{\mathbf{w}}, p, q \leftarrow 0$
while not converged do
    $p \leftarrow \mathrm{proj}_P\Big(p + \sigma(D\bar{u} - \bar{\mathbf{w}})\Big)$
    $q \leftarrow \mathrm{proj}_Q(q + \sigma\mathcal{E}\bar{\mathbf{w}})$
    $u_{\mathrm{old}} \leftarrow u$
    $u \leftarrow \mathrm{prox}_{\tau F}\Big(u + \tau(D^* p)\Big)$
    $\bar{u} \leftarrow 2u - u_{\mathrm{old}}$
    $\mathbf{w}_{\mathrm{old}} \leftarrow \mathbf{w}$
    $\mathbf{w} \leftarrow \mathbf{w} + \tau(p + \mathcal{E}^* q)$
    $\bar{\mathbf{w}} \leftarrow 2\mathbf{w} - \mathbf{w}_{\mathrm{old}}$
end while
return  u
```

Algorithm 2: TGV denoising algorithm

Now we can once again exploit that $F(u, \mathbf{w}) = \frac{1}{2}\|u - v\|_2^2$ is independent of $\mathbf{w}$. The optimal choice of $\hat{\mathbf{w}}$ is thus the one that minimizes the Euclidean distance to w, as the second term in (4.28) is the only one containing $\hat{\mathbf{w}}$. This choice is $\mathbf{w}$ itself, so the proximal point mapping on the second variable is simply the identity operator. We are now left with a minimization problem only for u.

$$\mathrm{prox}_{\tau F}(u, \mathbf{w}) = \left(\arg\min_{\hat{u}} \Big(\frac{1}{2}(\|\hat{u} - u\|_2^2 + \tau\|u - v\|_2^2)\Big), \mathbf{w}\right). \tag{4.29}$$

Inserting $u - \tau D^* p$ as the first variable, we obtain exactly the minimzation problem (3.12). We solved this problem for total variation, and by following the same procedure here we can conclude that

$$\mathrm{prox}_{\tau F}(u, \mathbf{w}) = \left(\frac{u - \tau D^* p + \tau v}{1 + \tau}, \mathbf{w}\right). \tag{4.30}$$

Duality gap for TGV

As for the case with single-parameter convex optimization, we apply the Fenchel-Rockafeller theorem and use the duality gap between the primal and dual functional as out convergence criteria. This duality gap gives an upper bound for distance to the optimal value of the objective function.

We can start by simply inserting the discretized primal and dual functionals from the optimization algorithm, to obtain an expression for the duality gap. From this expression, however, we will follow the idea presented in [KLV19] to simplify the duality gap somewhat.

$$\begin{aligned}\mathrm{gap}_{\mathtt{TGV}}(u, \mathbf{w}, p, q) = &\frac{1}{2}\|u - v\|_2^2 + \lambda_1\|\nabla u - \mathbf{w}\|_1 + \lambda_2\||\mathcal{E}\mathbf{w}\||_1 \\ &+ \frac{1}{2}\|\nabla^* p\|_2^2 - \langle p, \nabla v\rangle + \mathcal{I}_{\{0\}}(p - \mathcal{E}^* q).\end{aligned}$$

The simplification arises from the observation that if $p \neq \mathcal{E}^* q$, the last term will become plus infinity, and the duality gap itself would also be infinite. Therefore we require that $p = \mathcal{E}^*q$, and obtain a duality gap only depending on the primal variables and one dual variable q.

$$\mathrm{gap}_{\texttt{TGV}}(u, \mathbf{w}, q) = \frac{1}{2}\|u-v\|_2^2+\lambda_1\|\nabla u-\mathbf{w}\|_1+\lambda_2\||\mathcal{E}\mathbf{w}\||_1+\frac{1}{2}\|\nabla^*(\mathcal{E}^*q)\|_2^2-\langle\mathcal{E}^*q, \nabla v\rangle. \tag{4.31}$$

We discretize this duality gap to arrive at the following expression for the duality gap for TGV denoising:

$$\begin{aligned}\mathrm{gap}_{\texttt{TGV}}(u, \mathbf{w}, q) = \Big(&\frac{1}{2}\|u-v\|^2 + \lambda_1\sum_{i=1}^{n}\sum_{j=1}^{m}|\nabla u_{(i,j)} - \mathbf{w}_{(i,j)}| + \lambda_2\sum_{i=1}^{n}\sum_{j=1}^{m}|\mathcal{E}\mathbf{w}_{(i,j)}| \\ &+\frac{1}{2}\|D^*(\mathcal{E}^*q_{(i,j)}\|_2^2 - \sum_{i=1}^{n}\sum_{j=1}^{m}(\mathcal{E}q^{(1)}_{(i,j)}\nabla v^{(1)}_{(i,j)} + \mathcal{E}q^{(2)}_{(i,j)}\nabla v^{(2)}_{(i,j)})\Big).\end{aligned} \tag{4.32}$$

4.4 TGV as a single-parameter method

It is easier to perform single-parameter selection than multi-parameter selection. We can develop a TGV-based single-parameter method by defining a fixed relation $\gamma = \frac{\lambda_1}{\lambda_2}$ between the two regularization parameters. Then the vector $\boldsymbol{\lambda}$ of regularization parameters can be written as $\boldsymbol{\lambda} = [\lambda_1, \gamma\lambda_1]^T$ The algorithm is now exactly the same, but without the possibility of balancing the regularization terms against each other, as we only can vary λ_1. In this special case we can rewrite the regularization terms from Equation (4.3) as

$$\mathrm{TGV}(u, \mathbf{w}; \gamma) = \lambda\sum_{i=1}^{n}\sum_{j=1}^{m}|\nabla u_{i,j} - \mathbf{w}_{i,j}| + \gamma\lambda\sum_{i=1}^{n}\sum_{j=1}^{m}|\mathcal{E}\mathbf{w}_{i,j}|. \tag{4.33}$$

with λ as the only regularization parameter. The minimization problem does also rely on only one regularization parameter. This problem can be stated as the functional

$$T_{\boldsymbol{\lambda}}(v;\gamma) = \min_{u,\mathbf{w}}\Big[L(u,v) + \lambda\sum_{i=1}^{n}\sum_{j=1}^{m}|\nabla u_{i,j} - \mathbf{w}_{i,j}| + \gamma\lambda\sum_{i=1}^{n}\sum_{j=1}^{m}|\mathcal{E}\mathbf{w}_{i,j}|\Big]. \tag{4.34}$$

If one chooses a fixed γ, it is possible to treat the functional (4.34) as a single-parameter problem, and solve it numerically in a similar manner to the approach we use in the minimization problem for total variation that was defined in Equation (3.3). In the next section we will discuss numerical parameter choice algorithms that can be used on single-parameter problems of this type, in order to determine the optimal parameter choice λ for obtaining good reconstructions of the original images.

5 Numerical experiments with single-parameter choice algorithms

This section aims to give an overview of some of the more common parameter selection algorithms, and to show some examples where these algorithms are applied. The discrepancy rule and the L-curve method are introduced and implemented, and used to automate parameter choices. The Chambolle-Pock algorithm for total variation is used as a numerical solver. The results are compared with results for a quadratic regularization functional. All algorithms in this section are single-parameter optimization algorithms.

Choosing optimal or good regularization parameters for total variation in three RGB channels is far from trivial. In this section we will scale the range of the regularization parameter λ down to the unit interval in $\mathbb{R}^+$. Recalling Equation (3.3), we now consider it a parameter selection problem, in which the parameter $\lambda \in \mathbb{R}^+$ is the parameter that should be optimized. The set of all positive real numbers is a big domain to search through for an optimal parameter choice, so a scaling of the problem is helpful. Therefore a parameter $t \in [0, 1]$ is introduced, which is defined by $\lambda := \frac{1-t}{t}$. Equation (3.3) can then be rewritten as

$$S(t)(v) = \arg\min_{u \in U} \frac{t}{2}\|u - v\|^2 + (1-t)\mathrm{TV}(u). \tag{5.1}$$

And this is the equation that we will use for the remainder of this section, where two different numerical algorithms for choosing the optimal parameter t are investigated. The discrepancy principle, which is useful when the noise level is known, and the L-curve rule which is applied in cases where the noise level is unknown to us.

5.1 Discrepancy principle

The discrepancy principle is a parameter choice method in which the expected noise level $\sigma > 0$ is known. For a discrete image denoising problem, this can be written exactly in the form of (2.2) with σ as the noise level. Assume that the known noisy data or measurement is given as $v \in \mathbb{R}^{3mn}$. We recall that v is generated from a Gaussian distribution with standard deviation σ. But because of the randomness of the method, the actual realized noise level will (most likely) not be exactly σ. Let moreover the numerical solution of the denoising problem, given the parameter choice λ, be denoted by u_λ. The idea of the discrepancy principle is then to require the solution to be such that $\|u_\lambda - v\| \approx \sigma$. This means that the residual and the expected noise is of approximately the same size.

In order to produce a more stable version of the discrepancy principle, one may require $\|u_\lambda - v\| = C\sigma$ for a $C > 1$. In the numerical experiments to be presented, however, we always use $C = 1$. This gives the following formulation of the minimization problem given by the discrepancy principle in the general case:

$$u_\lambda = \arg\min_u \Big[R(u) \text{ such that } \|u - v\|_2^2 = C\sigma \Big]. \tag{5.2}$$

In the case of total variation, where the regularization term is $\mathrm{TV}(u)$, we obtain

$$u_\lambda = \arg\min_u \Big[\mathrm{TV}(u) \text{ such that } \|u - v\|_2^2 = C\sigma \Big]. \tag{5.3}$$

Existence of solutions

When applying the discrepancy principle, it is not guaranteed that solutions exist for all noise levels. In this subsection, we will develop criteria for when a solution exist, and discuss its properties. We will see that if the noise level is too high, the discrepancy principle will not result in solutions to the denoising problem. We recall the functional we want to minimize in the denoising problem, given a general regularization functional $R(u)$.

$$\frac{1}{2}\|u-v\|_2^2 + \lambda R(u). \tag{5.4}$$

The idea of the discrepancy principle is to rewrite this functional in Equation (5.4) such that we require the first term to be equal to $\frac{1}{2}C^2\sigma^2$, and then minimize over the second term. Moreover, to ensure a convex constraint, we introduce a convex relaxation and change this equality constraint to an inequality constraint. This results in the following (still assumed to be finite-dimensional) minimization problem:

$$\min_u R(u) \text{ such that } \|u-v\|^2 \leq C^2\sigma^2. \tag{5.5}$$

In the finite dimensional case, this is a convex problem if the regularization functional is convex. For total variation denoising, $R(u)$ is both lower-semi-continuous and coercive for non-constant u, while the constraint is both convex and closed. Thus, a solution to the minimization problem (5.5) does exist for all realistic noise levels, as u is non-constant for low noise levels. We can consider the Lagrangian of the optimization problem, in order to establish KKT-conditions. The Lagrangian of this problem is given as

$$\mathcal{L}(u,\gamma) = \Big[R(u) - \frac{C\gamma}{2}\Big(\sigma^2 - \|u-v\|^2\Big)\Big], \tag{5.6}$$

with γ as the Lagrangian parameter. We want to minimize the Lagrangian in order to find the pair $(\bar{u},\bar{\gamma})$ such that $\bar{u}$ is the solution to (5.3). We do this by developing KKT-conditions from the Lagrangian itself. The first condition is that $\bar{u}$ minimizes the Lagrangian given a fixed $\bar{\gamma}$, and can be written as

$$\bar{u} = \arg\min_u \Big[R(u) - \frac{C\bar{\gamma}}{2}\Big(\sigma^2 - \|u-v\|^2\Big)\Big]. \tag{5.7}$$

By removing the term $\frac{\gamma\sigma^2}{2}$, which is independent of u, this minimization problem can be further simplified to

$$\bar{u} = \arg\min_u \Big(R(u) + \frac{C\bar{\gamma}}{2}\|u-v\|^2\Big). \tag{5.8}$$

The other KKT-conditions state that the Lagrangian parameter must be non-negative, that the constraint must be fullfilled, and that in the case of an strict inequality in the constraint, we have that $\bar{\gamma} = 0$. Thus, the system of KKT conditions can be written as

$$\bar{u} = \arg\min_u \left(R(u) + \frac{C\bar{\gamma}}{2}\|u - v\|^2 \right) \tag{5.9}$$

$$\|\bar{u} - v\| \leq C\sigma \tag{5.10}$$

$$\bar{\gamma} \geq 0 \tag{5.11}$$

$$\bar{\gamma} = 0 \text{ if } \|\bar{u} - v\| < C\sigma. \tag{5.12}$$

We now consider two different cases for the inequality constraints. In the case where $\|\bar{u}-v\| < C\sigma$, we obtain $0 \in \partial R(u)$ and $\bar{\gamma} = 0$. This means that the discrete signal is constant over the entire domain. A feasible constant solution could be the constant function whose value is the average of v. Any constant function $k \in \mathbb{R}^{3mn}$ such that $\|k - v\| < C\sigma$ will solve (5.8) in the situation with a strict inequality in the constraint on the norm. A constant solution can thus only occur when the variance of the noisy image is smaller than the constant C^2 [WC12]. For the noisy image v, we would expect the variance to be slightly smaller than $\text{Var}(u) + \sigma^2$. In the case of an strict inequality constraint, we thus have either that σ is very high, or that the true solution u has (almost) zero variance. Constant solutions will thus only occur for high noise levels.

We know that if $\|u_\lambda - v\| < C\sigma$ for all $\lambda \in \mathbb{R}^+$, that means that noise level (multiplied by C) will be bigger than the difference between the noisy and reconstructed parameter for all parameter choices. It is in this case the first constraint has a strict inequality, and the discrepancy principle fails to produce a parameter choice for λ. Numerical experimentation shows that this occur for high noise levels for the KODAK barn image. We also recall that the possibility of an inequality was introduced as a convex relaxation to ensure convexity. Thus, a constant solution will not solve the original minimization problem from Equation (3.1).

When equality holds in the inequality constraint, we have that $\gamma \geq 0$, and more importantly that $\|\bar{u}-v\| = C\sigma$. In this case, a solution exists if there is a parameter $\lambda \in \mathbb{R}^+$ which produces $\bar{u}$ as the reconstructed solution using total variation. This parameter then is the solution to the original total variation minimization problem from Equation (3.1).

Numerical implementation of the discrepancy principle

Recalling that digital images can be considered as digital signals of dimension $n \times m$, we turn to an error estimate of the discrepancy principle. In cases where the noise level σ is known, the size of the error is given as $C\sigma\sqrt{nm}$. In the situation we are dealing with, where the actual noise is a realization of an independently and individually distributed Gaussian variable, the actual noise level will be distributed according to a chi-squared distribution with $3nm$ degrees of freedom. We will be using $C\sigma\sqrt{nm}$ as our error tolerance in the numerical implentation. It is common to let $C = 1$, and we will follow this convention in all numerical experiments. [WC12; GK92].

While searching for good parameter choices t, the following procedure is implemented, and described in Algorithm 3:

We can see that the idea is to increase t gradually, until we reach a t such that $\hat{u}(t)$ satisfies the convergence criteria. The constants q and r determines how t is updated, and how large steps that are performed. The following series of t_i-s are formed:

```
Initialize $i = 0, q = 0.9, r = 1$
$\texttt{TOL} = C\sigma\sqrt{nm}$
while $\|\hat{u}(t) - v\| > \texttt{TOL}$ do
    $t_i \leftarrow \frac{1}{1+rq^i}$
    $\lambda \leftarrow \frac{1-t_i}{t_i}$
    $\hat{u}(t_i) \leftarrow \textsf{chambolle-pock}(v, \lambda)$
    $i \leftarrow i + 1$
end while
return $\hat{u}$
```

Algorithm 3: Discrepancy rule algorithm

$$[t_0, t_1, t_2, \ldots, t_k]^T = [\frac{1}{1+r}, \frac{1}{1+rq}, \frac{1}{1+rq^2}, \ldots, \frac{1}{1+rq^k}]^T.$$

With our choices of r and q, we start with $t_0 = 0.5$, and keep on choosing higher values as long as $\|\hat{u}(t) - v\| > \texttt{TOL}$. The steps become shorter as t grows. The advantage of this will be visible in the figure below.

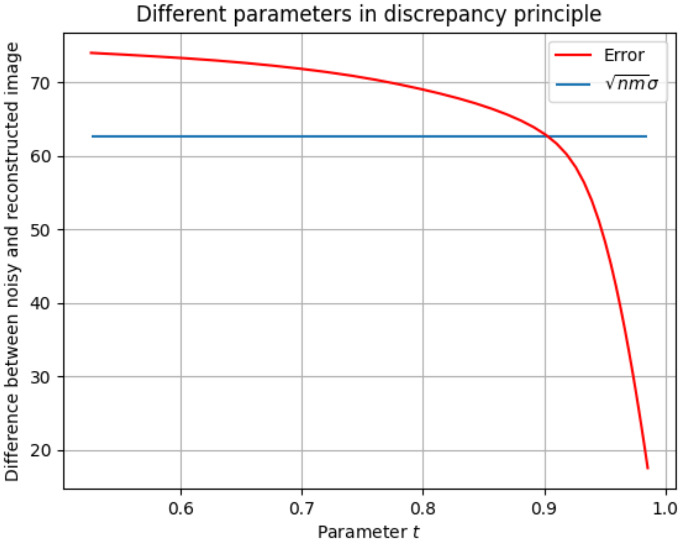
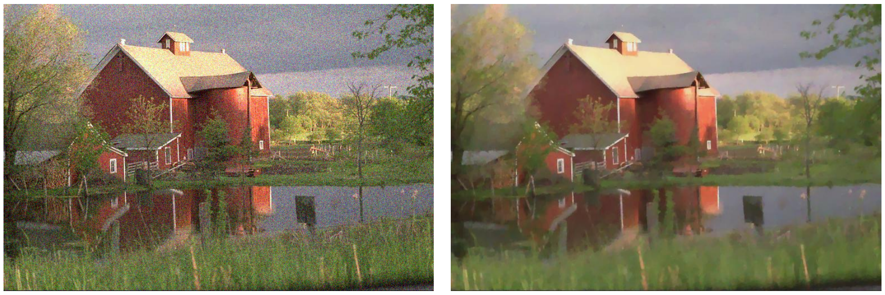


Figure 5.1: An example of errors with different parameter choices for the noise level $\sigma = 0.1$

Figure 5.1 shows a plot of the Frobenius norm of the difference between the reconstructed solution $\hat{u}$ and the noisy solution v for different values of the parameter t. We see that another advantage of the scaling is that we get to restrict the length of the first axis, and that it becomes easier to visualize results. The horizontal line is the upper bound for the error $(C\sigma\sqrt{nm})$, and we observe that higher values of t produce smaller errors. Choices of t close to the blue line will be a 'good choice' according to the discrepancy principle, and the algorithm is implemented such that it chooses the first t that has $\|\hat{u}(t) - v\| < C\sigma\sqrt{nm}$.

The noisy image, still with noise level $\sigma = 0.1$, has an PSNR score of 20.10 when compared to the original noise-free image. The reconstructed image, produced by the discrepancy rule parameter choice algorithm has a PSNR value of 28.01. The two images are presented in Figure 5.2. We observe that the reconstruction is less noisy and more smooth than the noisy image. However, we do also observe that the reconstruction suffers from the main weakness of total variation, with piecewise constant domains in the image.

Figure 5.2: Noisy and reconstructed image, using discrepancy principle.

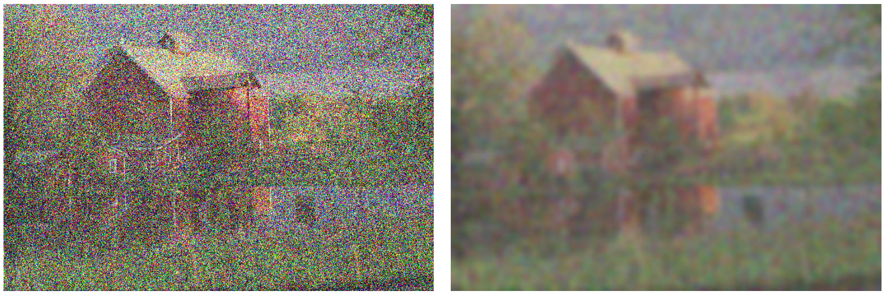


(a) Noisy image, $\sigma = 0.1$.

(b) Reconstructed image.

We also include an example at a higher noise level, where the discrepancy principle is known not to be too efficient[BM12]. The noisy image with noise level 0.65 and the reconstruction using the discrepancy principle is compared in Figure 5.3. We can see that the reconstruction is much smoother than the noisy image. The barn is still visible in both images, but is not very clear in either of them. The PSNR ratio of the reconstruction is three times larger than for the noisy image, when compared to the original KODAK image.

Figure 5.3: Noisy and reconstructed image, using TV discrepancy principle.$\sigma = 0.65$.



(a) Noisy image, $\sigma = 0.65$.

(b) Reconstructed image.

5.2 L-curve method

Another well known method for image denoising is the L-curve method. This method is used in cases where the noise level is not known. The essence of the L-curve method is to plot the regularization term and the residuals against each other on a appropriate scale for different choices of the regularization parameter. In the single parameter-setting we would obtain a curve in two dimensions. If there are more than one regularization term and more than one regularization parameter, the plot will be a hypersurface in $M + 1$ dimensions, where M is the number of regularization terms. In this subsection we consider an L-curve approach for total variation, which is a single-parameter method. Logarithmic scales are commonly used, and will be used in all L-curve visualizations in this subsection. The visualizations of the two terms give a visual insight into where the two terms of the cost functions are reasonably balanced. We will see that the plot often is L-shaped, and that choosing the regularization parameter for which the curvature of the curve is highest, will be a viable approach.

We again consider a denoising problem for discrete signals, and a general regularization functional $R(u)$. We have some fixed $v \in V$, which is a 'noisy' solution or measurement to the problem $\arg\min_v \|u - v\|_2^2$. Introducing a regularization parameter λ, we can express the problem as a minimization problem

$$u_\alpha = \text{argmin}_u \left[\frac{1}{2} \|Au - v\|_2^2 + \lambda R(u) \right] \tag{5.13}$$

for some regularization term $R(u)$. As previously discussed, there are different functionals that may serve as regularization functionals. For now however, we will let $R(u)$ be a general regularization term.

Note that when $\lambda \longrightarrow 0$, we will have that $\|Au - v\|_2^2 \longrightarrow 0$ as well. A decrease in λ is in other words the same as regularizing less. On the other hand, if $\lambda \longrightarrow \infty$, the minimization will mostly be concerned about minimizing the second term. In the case where $R(u) = \|u\|_2^2$, the consequence would be that $\|u\|_2^2 \longrightarrow 0$, and thus $\|Au - v\|_2^2 \longrightarrow \|0 - v\|_2^2 = \|v\|_2^2$. In the case of total variation, that is $R(u) = \text{TV}(u)$, what will happen if $\lambda \longrightarrow \infty$? We would then have that all edges are heavily penalized, and the (over-)regularized solution u would become constant. We would also observe the same for a quadratic regularization term, when $\lambda \longrightarrow \infty$.

The L-curve method exploits this, and states that the best choice of the regularization parameter λ must be somewhere in between the under- and over-regularized solution. This is where a loglog-plot turns out to be useful. One plots the logarithm of the regularization term versus the logarithm of the norm of the residual, and will normally obtain a L-shaped curve. Typically, one will choose parameters in the corner or 'elbow' of the L-shaped curve, which often looks like the sketch in Figure 5.4. It has previously been shown, see for example [HO93], that a plot of the Euclidean norms of the discrepancy term versus the regularization term is a good starting point for the L-curve method. From this plot, one can start the visual inspection, or numerically calculate the Gaussian curvature of the resulting curve.

When choosing parameters in the corner or 'elbow' of the L-curve, one has by graphical tools chosen a parameter for which the regularized solution is not too dominated by neither the residual, nor the regularization term. In [HO93], the authors suggests two useful possible implementations of a L-curve method. One may

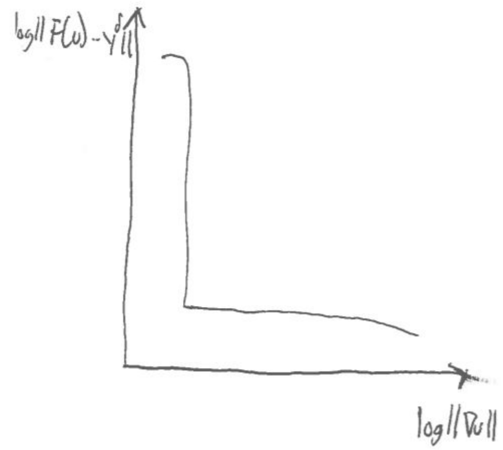


Figure 5.4: A sketch of L-shaped curve illustrating trade-off between residual and regularization term.

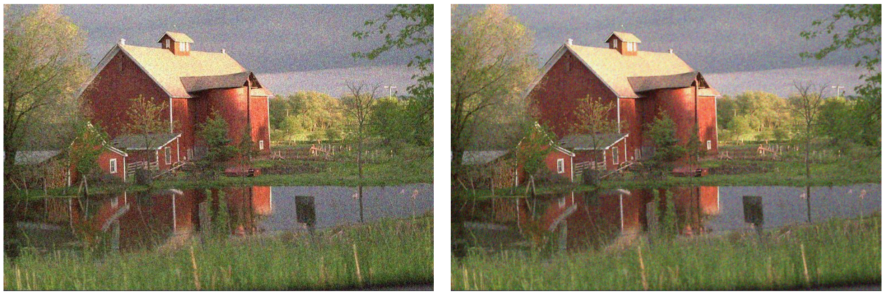
either choose the point on the curve with the shortest distance to the origin, or one may choose the point on the curve with the highest maximum curvature. Moreover, they warn that the first approach may cause somewhat under-regularized solutions, due to effects of different perturbation errors. In the numerical implementation, I therefore use the latter approach, and choose the point on the curve with the maximum curvature (in absolute value).

Regarding the numerical implementation of the L-curve, it has been performed by the same principles as for the previous methods. That is, the three RGB-channels have been treated separately, and parameter selection has been performed channel-wise. The same procedure as for the discrepancy principle has been implemented, where we have $t_i = \frac{1}{t_0 q^i}$ and $\lambda_i = \frac{1-t_i}{t_i}$ for the i-th parameter being sent to the Chambolle-Pock solver.

For $\sigma = 0.1$, a noisy and reconstructed image (using the L-curve method) is presented in Figure 5.5. We can see with the naked eye that the reconstruction is much less smooth than the reconstruction from the discrepancy principle, that was presented in Figure 5.2.

The L-curves produced by the experiment do however not have the L-shaped form we recognise from Figure 5.4. Instead we observe that we have an L-shape with the elbow of the L-curve in the opposite corner of the plot, see Figure 5.6. This means that there are areas where both the residual and the regularization term have high values. However, it is still a valid approach in this setting to use the point with the maximum curvature when performing the parameter selection[JG00]. Therefore, the procedure we use when determining regularization parameters takes the following form: We plot the logarithm of the residual versus the logarithm of the regularization term. Then we calculate the Gaussian curvature of the curve, and identify the point in which this curvature is maximized. The regularization parameter that was used in that point is then chosen to by the algorithm to be the optimal parameter. It is worth stressing that the approach of choosing the point with the maximum curvature is not the standard approach,

Figure 5.5: Noisy and reconstructed image, using L-curve method



(a) Noisy image, $\sigma = 0.1$.

(b) Reconstructed image.

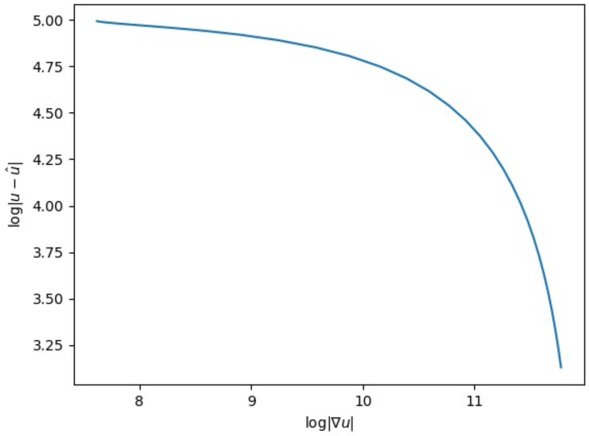


Figure 5.6: L-curve for total variation with different parameters

but as shown in [JG00], this is a viable approach in the setting where the L-cure looks like the one in Figure 5.6.

The typical result for the different noise levels that have been tested is that the parameter t chosen by the L-curve is slightly lower than the one chosen by the discrepancy principle for low noise levels.

5.3 Numerical experiments with single-parameter TGV

We recall from Section 4.4 that we may consider total generalized variation as a single-parameter regularization functional if we choose a fixed relationship γ between the regularization parameters. We derived the minimization problem (4.34), which is

$$T_\lambda(v;\gamma) = \min_{u,\mathbf{w}} \Big[L(u,v) + \lambda \sum_{i=1}^{n} \sum_{j=1}^{m} |\nabla u_{i,j} - \mathbf{w}_{i,j}| + \gamma\lambda \sum_{i=1}^{n} \sum_{j=1}^{m} |\mathcal{E}\mathbf{w}_{i,j}| \Big].$$

We want to investigate if using the discrepancy principle as a parameter selection rule will give good reconstructions of the original image. We reuse the algorithm from Section 5.1, which chooses the first λ such that $\|u_\lambda - v\|_2 \leq C\sigma$. The algorithm is initialized in the same way, and we let C remain equal to 1. Different values for γ have been tested and compared, and based on the results from the tests we use $\gamma = \frac{1}{2}$ in the main numerical experiment for single-parameter methods.

5.4 Comparison of single-parameter methods

An interesting question is of course which of the denoising methods that produce the best results and the most accurate reconstructions of the original image. We use the PSNR metric to compare the quality of reconstructions created by the discrepancy rule, the L-curve method and the single-parameter approach for total generalized variation.

As a baseline for our comparison we use the results from quadratic regularization from before. The results from quadratic regularization have been created by using the conjugate gradient method (see Appendix A) with tolerance 10^{-7} for different parameter choices $\lambda \in [0, 5]$, and then choosing the reconstruction with the highest PSNR score. The quadratic regularization reconstruction is thus not the result of a heuristic parameter choice algorithm, unlike the other methods.

The different parameter choice methods have been tested for the uniformly spaced noise levels $0.1, 0.2, 0.3, 0.4$ and 0.5. On each noise level, we have created ten realizations of a noisy image with the given noise levels. These images have been used as input to the different denoising algorithms. The average PSNR scores of the different methods at the different noise levels are presented in Figure 5.7. For the discrepancy principle and the L-curve method, the previously explained (and fully automated) parameter search algorithms for total variation denoising is applied. The discrepancy rule is also used for single-parameter TGV.

The numerical experiments show that the L-curve method performs worse than both quadratic regularization and the discrepancy principle for a broad range of noise levels, at least in the PSNR metric. See Figure 5.7 for the results. We can see that the reconstructed images using an L-curve parameter have higher PSNR values than the noisy images, but lower values than those reconstructed with other means. This makes sense, as we in the discrepancy principle have knowledge about the actual noise level. This knowledge is not present in the L-curve method. We would thus expect the discrepancy principle to perform better than the L-curve method.

When we compare the results from the discrepancy principle from the two different regularization functionals, we observe that TGV produce significantly better results for $\sigma = 0.2$, and that TV and TGV essentially perform equally well for most other noise levels. For $\sigma = 0.1$, both these methods generate better reconstructions than quadratic regularization, whereas it is harder to distinguish between the reconstructions at higher noise levels. Comparisons performed in the SSIM metric do not add any new information, as the graphs follow the same pattern as for total variation.

We have performed a closer investigation and a statistical analysis of the results at noise level $\sigma = 0.1$. The box plot in Figure 5.8 shows the results from each of the ten realizations. The plot is zoomed in at the three best-performing methods, to illustrate the differences in their performance. The methods that are tested are quadratic regularization, the discrepancy rule for total variation (DR) and the

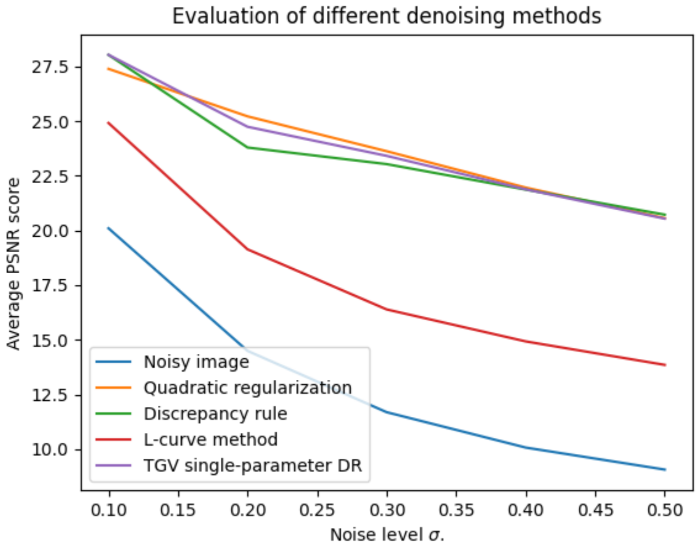


Figure 5.7: PSNR values for different methods at 61 different and uniformly spaced noise levels.

discrepancy rule for single-parameter total generalized variation (TGV) with $\gamma = \frac{1}{2}$. It is clear from the box plot that total generalized variation beats total variation for all ten realizations, measured in the PSNR metric. The statistical analysis clearly shows that the variance in each method is much smaller than the differences between the methods. Thus, we can be reasonably sure that the parameter choice method is consistent and that our evaluation measure actually measures the quality of the methods, and not random effects caused by noise or statistical errors. In Section 7 we will take a closer look at the visual differences between reconstructions originating from total variation and total generalized variation respectively, but first we will introduce the concept of multi-parameter choice methods.

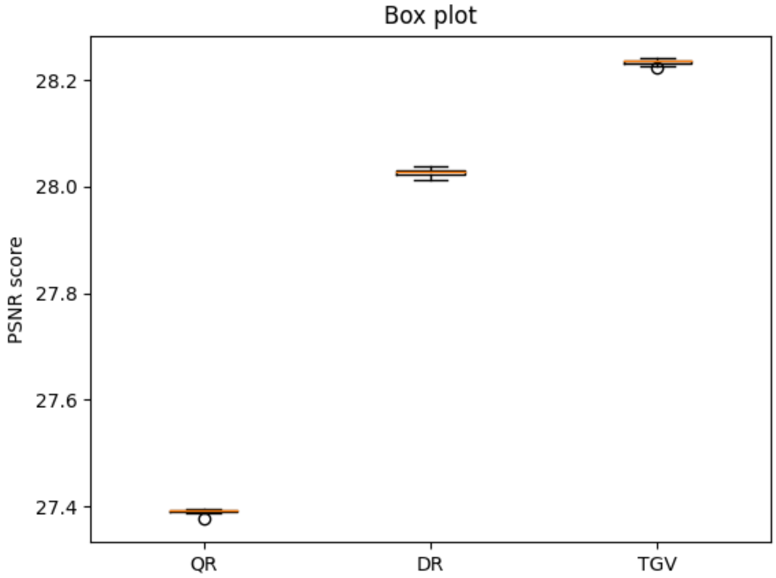


Figure 5.8: Box plot with denoising results for ten realizations with $\sigma = 0.1$. Notice the truncated y-axis.

6 Multi-parameter choice methods

Choosing good parameters is essential to most optimization algorithms. Runtime, efficiency and even the feasability of the numercial solutions may depend on the choice of such parameters. We have previously considered some parameter choice methods for single-parameter optimization. In this section, we will consider the more complicated case of multi-parameter optimization, primarily for total generalized variation. Multi-parameter optimization allows multiple penalty terms to be included in the cost function, and the choice of parameters decides how the different penalty terms should be weighted against each other. Having multiple parameters increases the flexibility of the method and may allow for more accurate penalizations of multiple propertites of a function, but does also complicate both analytical and numerical calculations to some extent.

Popular multi-parameter approaches are often extensions of one-dimensional parameter choice methods, which need to be modified, changed or made more efficient before being applied in the more complex are of multi-parameter selection. In this section, we are going to consider and derive Broyden's algorithm, a quasi-Newton algorithm we will use as a numerical solver for the discrepancy principle. Moreover, we will see that a L-curve can be generalized into more dimensions by a L-hypersurface. We will also derive a fixed point iteration based on an L-hypersurface, that can be used to perform multi-parameter selection without knowledge about the noise level of the image.

In a general Tikhonov regularization setting for image denoising, we consider a regularization functional with N regularization terms, which can be expressed as functions of the reconstruction u. We denote these functions by $\psi_1(u), \psi_2(u), \ldots, \psi_N(u)$. In the case of total generalized variation, we know that we have two regularization terms ψ_1 and ψ_2, with regularization parameters λ_1 and λ_2.

Denote by $\boldsymbol{\lambda} := (\lambda_1, \lambda_2, \ldots, \lambda_N)^T \in \mathbb{R}^N_{(+)}$ the vector containing all N regularization parameters. As for total variation, the choice $\boldsymbol{\lambda} = \mathbf{0}$ will mean that no regularization is performed, and that the Tikhonov regularization problem is reduced to a least squares problem.

6.1 Discrepancy principle for multi-parameter optimization

As in the case with single-parameter optimization, the discrepancy principle for multiple parameters is applied in situations where we know the expected noise level σ, and we also assume that the operator A is both linear and injective. We follow the notation presented in [LP11], where the following multi-parameter functional is suggested:

$$\hat{u} = \arg\min u \|Au - v\|_2^2 + \sum_{i=1}^{N} \lambda_i \psi_i(u). \tag{6.1}$$

The authors also suggest to include a penalty term $\beta\|u\|_2^2$, with $\beta > 0$ as a regularization parameter which weighs the norm of the image u[LP11]. We will not include this term in our model, mainly for two reasons. Firstly, the term $\|u\|_2^2$ is a function of u, and could thus be included in the general model as one of the $\psi_i(u)$-terms, if needed. More important, an inclusion of this term would not be helpful in the models that will be described later. This extra regularization term will essentially penalize high values in the RGB-channels, a penalty which is not desired for our denoising purposes. This penalty would in practice penalize all images that are not entirely black (0 in all RGB-channels). For reconstruction of images with many and bright colours, this penalty is therefore not a relevant or desirable one to include in a denoising functional.

As for the one-dimensional discrepancy principle, a constant $c \geq 1$ is introduced, and the generalized discrepancy principle takes the form

$$\|A\hat{u}(\boldsymbol{\lambda}, \beta) - v\| = c\sigma. \tag{6.2}$$

Similar to the calculations we preformed for total variation, we let $c = 1$ in all further calculations. It is common to use a slightly higher value for c in cases where one needs to obtain a more stable algorithm.

We denote by $u_{\boldsymbol{\lambda}}^{\sigma}$ the regularized and denoised numerical solution at noise level σ with regularization parameters $\boldsymbol{\lambda}$. From the discrepancy principle given in Equation (6.2), we can expect that $\|u_{\boldsymbol{\lambda}}^{\sigma} - v\| \approx \sigma$. The calculations of the KKT conditions that were performed in the last section are valid for a general regularization term $R(u)$, as long as the regularization term is lower semi-continuous and coercive.

In the case of total generalized variation, we see that the norm of the symmetrized gradient goes to infinity if w does, and moreover the norm of $\|\nabla u - w\|$ also does. From this we conclude that the regularization terms are coercive. They are also continuous, and thus they are lower semi-continuous, and the calculations from last section are still valid.

Moreover, we will apply a method known as **the balanced discepancy principle**. This name is given because the method requires each of the regularization terms to contribute equally to the regularization. That is, we need to choose parameters $\boldsymbol{\lambda}$ such that $\lambda_i \psi_i(u) \approx \lambda_j \psi_j(u)$ for all i and j. The idea of this is to ensure that all the undesired properties are penalized almost equally. This is not always optimal, but is used as a method to design and add another equation to the system that shall be solved. When we optimize for more than one parameter, we need a system of equations that contains more than the one equation given in Equation (6.2). Using the balanced discrepancy principle on our total generalized variation functional gives two equations and two unknowns, and thus the system of equations may be solved. We will now introduce a numerical method for solving systems of equations, namely the Broyden's method.

6.1.1 Broyden's method

Introduced by C.G Broyden in 1965, Broyden's method is a numerical method for solving multiple non-linear equations simultaneously [Bro65]. In the method, the zeros of each equation are searched for by performing a line search. This is similar to the idea we know from the more famous Newton's method, and the new method is an extension, or even an improvement, of the Newton method[Bro65].

Broyden's method is a quasi-Newton method. Similar to the Newton method, quasi-Newton methods can be used to maximize or minimize functions, or to identify zeros. It is the latter approach that is applied in Broyden's method. However, quasi-Newton methods are different from Newton methods in some essential ways. It is often computationally expensive to calculate the Jacobian (or even the Hessian) exactly. Quasi-Newton methods cope with this by not calculating these matrices explicitly in each numerical iteration. Instead, an approximation to the matrix is estimated, and this approximation is used in the next updates[AK07].

Consider a function $F : \mathbb{R}^m \to \mathbb{R}^n$. The input variables to the function can be denoted $\mathbf{x} = (x_1, x_2, \ldots, x_m)^T$. When applying a quasi-Newton method on the function, we can denote the vector of input variables to the function in iteration k by $\mathbf{x}_k$. In Broyden's method, we want to find the zeros of F, that is, we want to find the vector $\mathbf{x}$ such that $F(\mathbf{x}) = \mathbf{0} \in \mathbb{R}^n$.

We can now continue by defining

$$s_k = \mathbf{x}_{k+1} - \mathbf{x}_k$$

as the performed step in each iteration. The step length is given as the square root of the Euclidean inner product of s_k with itself, $\sqrt{s_k^T s_k}$. Similarly, we let $Y_k = F(\mathbf{x}_{k+1}) - F(\mathbf{x}_k) \in \mathbb{R}^n$ be the vector which represents the change in the function values in the n different components since the previous iteration.

In order to obtain better updates, we need approximations to the Jacobian matrices. We denote the approximation to the Jacobi matrix J_k in iteration k by $\mathcal{J}_k$. The update $\mathcal{J}_{k+1}$ depends on $\mathcal{J}_k$, and is given by Nocedal and Wright [NW06] as

$$\mathcal{J}_{k+1} = \mathcal{J}_k + \frac{(Y_k - \mathcal{J}_k s_k) s_k^T}{\|s_k\|^2}. \tag{6.3}$$

When applying Broyden's method, the update makes the smallest possible change (measured in the Frobenius norm) to the approximation of the Jacobian matrix, under the requirement that the secant equation $\mathcal{J}_{k+1} s_k = Y_k$ still has to be fullfilled.[NW06] Dennis and Schnabel [DS83] has shown that the choice of $\mathcal{J}_{k+1}$ given in Equation (6.3) is the choice that satisfies this criteria. We will also prove this, by following the main idea and approach in the proof given on page 280 in [NW06]. In the proof, we will need the result of the following lemma.

Lemma 6.1. *For any vector* $s \in \mathbb{R}^n$, $\left\| \frac{ss^T}{s^T s} \right\| = 1$.

Proof. Let $s \in \mathbb{R}^n$.Consider first the eigenvalues of the matrix $\frac{ss^T}{s^T s}$. We observe that s is an eigenvector to the matrix with corresponding eigenvalue 1, as $\frac{ss^T s}{s^T s} = s$. Moreover, we know that the orthogonal complement of s is a set Θ in $\mathbb{R}^{n-1}$. Thus, for any basis vector $\theta \in \Theta$, we have that $s^T \theta = 0$ and therefore also that $\frac{ss^T \theta}{s^T s} = 0$. So θ is an eigenvector with eigenvalue 0. This means that the eigenspace corresponding to the eigenvalue 0 has dimension $n-1$, and thus the eigenvalue 0 has multiplicity $n-1$. This means that the only non-zero eigenvalue is 1.

We can now use the diagonalize the matrix $\frac{ss^T}{s^T s}$ by its orthogonal eigenvectors and the matrix Λ which contains the corresponsing eigenvalues. That is, $\frac{ss^T}{s^T s} = Q \Lambda Q^T$, for $\Lambda = \operatorname{diag}(1, 0, 0, 0 \ldots, 0)$. From linear algebra we have that the Frobenius norm of a matrix is equal to the trace of the matrix multiplied with its transpose. Thus we can write

$$\|Q \Lambda Q^T\|_F = \operatorname{Tr}\Big(Q \Lambda Q^T (Q \Lambda Q^T)^T\Big).$$

We can exploit that Q is orthogonal and that $Q^T Q = Q Q^T = I$, and write

$$\|Q \Lambda Q^T\|_F = \operatorname{Tr}\Big(Q \Lambda (Q^T Q) \Lambda^T Q^T\Big) = \operatorname{Tr}\Big(Q \Lambda \Lambda^T Q^T\Big).$$

Now, since Λ is a diagonal matrix, then $\Lambda\Lambda^T = \Lambda^2$. Moreover, we have from linear algebra that $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$ for general matrices A and B. Therefore, we obtain

$$\|Q\Lambda Q^T\|_F = \mathrm{Tr}(Q\Lambda^2 Q^T) = \mathrm{Tr}(\Lambda^2 Q^T Q),$$

where we again can exploit that $Q^T Q = I$. Thus we have that

$$\|Q\Lambda Q^T\|_F = \mathrm{Tr}(\Lambda^2),$$

which we know is 1.

□

Theorem 6.1. The matrix $\mathcal{J}_{k+1}$ defined in Equation (6.3) is the matrix that minimizes $\|M - \mathcal{J}_k\|$, subject to the constraint that it satisfies the secant equation $Ms_k = y_k$.

Proof. We start by rewriting Equation (6.3) to

$$\|\mathcal{J}_{k+1} - \mathcal{J}_k\| = \left\| \frac{(y_k - \mathcal{J}_k s_k)s_k^T}{\|s_k\|^2} \right\|.$$

Now let M be any matrix satisfying the secant equation $Ms_k = y_k$.

$$\|\mathcal{J}_{k+1} - \mathcal{J}_k\| = \left\| \frac{(Ms_k - \mathcal{J}_k s_k)s_k^T}{\|s_k\|^2} \right\|.$$

We now have a common factor s_k in both terms in the numerator, and can rewrite the expression as

$$\|\mathcal{J}_{k+1} - \mathcal{J}_k\| = \left\| \frac{(M - \mathcal{J}_k)s_k s_k^T}{s_k^T s_k} \right\|.$$

By the triangle inequality for norms, the equation can be reformulated as an inequality:

$$\|\mathcal{J}_{k+1} - \mathcal{J}_k\| \leq \|M - \mathcal{J}_k\| \left\| \frac{s_k s_k^T}{s_k^T s_k} \right\|.$$

By Lemma 6.1, the last of these norm is equal to 1, and we obtain

$$\|\mathcal{J}_{k+1} - \mathcal{J}_k\| \leq \|M - \mathcal{J}_k\|.$$

Since M is an arbitrary matrix satisfying the secant equation, we can conclude that $\mathcal{J}_{k+1}$ is the matrix that minimizes the distance, in the Frobenius norm, to $\mathcal{J}_k$. □

We can rewrite the (balanced) discrepancy principle for total generalized variation as

$$\mathbf{Y} = \begin{bmatrix} \frac{1}{2}\|u_{\boldsymbol{\lambda}}^{\sigma} - v\|_2^2 - \frac{1}{2}\sigma^2 + \lambda_2\psi_2(\mathbf{w}) - \lambda_1\psi_1(u, \mathbf{w}) \\ \frac{1}{2}\|u_{\boldsymbol{\lambda}}^{\sigma} - v\|_2^2 - \frac{1}{2}\sigma^2 + \lambda_1\psi_1(u, \mathbf{w}) - \lambda_2\psi_2(\mathbf{w}) \end{bmatrix}. \tag{6.4}$$

It is a straightforward calculation to determine the Jacobian J of $\mathbf{Y}$ with respect to $\boldsymbol{\lambda}$. This Jacobian is given as

$$J = \begin{bmatrix} \frac{\partial Y_1}{\partial \lambda_1} & \frac{\partial Y_1}{\partial \lambda_2} \\ \frac{\partial Y_2}{\partial \lambda_1} & \frac{\partial Y_2}{\partial \lambda_2} \end{bmatrix},$$

and thus the exact Jacobian is

$$J = \begin{bmatrix} -\psi_1(u, \mathbf{w}) & \psi_2(\mathbf{w}) \\ \psi_1(u, \mathbf{w}) & -\psi_2(\mathbf{w}) \end{bmatrix}. \tag{6.5}$$

Then Broyden's algorithm takes the following form for our multi-parameter choice problem.

```
Initialize $k = 0$ and choose $\boldsymbol{\lambda}_0 \in \mathbb{R}^2_+$.
$u_0, \mathbf{w}_0 \leftarrow$ TGV_DENOISING$(v, \boldsymbol{\lambda}_\mathbf{0})$
$J_0 \leftarrow$ JACOBIAN$(u_0, \mathbf{w}_0)$
$Y_0 \leftarrow Y(u_0, \mathbf{w}_\mathbf{0}, \boldsymbol{\lambda}_0)$
$\mathcal{J}_0 \leftarrow J_0$
for $k$ in range $(1, K)$ do
    $s_k^{\text{sugg}} \leftarrow -\mathcal{J}_{k-1}^{-1} Y_{k-1}$
    $s_k \leftarrow$ STEPCONTROL$(s_k^{\text{sugg}})$
    $\boldsymbol{\lambda}_k \leftarrow \boldsymbol{\lambda}_{k-1} + s_k$
    $u_k, \mathbf{w}_k \leftarrow$ TGV_DENOISING$(u_{k-1}, \boldsymbol{\lambda}_\mathbf{k})$
    $Y_k \leftarrow Y(u_k, \mathbf{w}_k, \boldsymbol{\lambda}_k)$
    $y_k \leftarrow Y_k - Y_{k-1}$
    $\mathcal{J}_k \leftarrow \mathcal{J}_{k-1} + \frac{(y_k - \mathcal{J}_{k-1} s_k) s_k^T}{\|s_k\|^2}$
end for
return $u_0, u_1, \ldots, u_K$
```

Algorithm 4: Broyden's method for TGV parameter choice

We only calculate the exact Jacobian matrix for the intital parameter choice $\boldsymbol{\lambda}_0$, and use the Broyden iterations to update our estimates $\mathcal{J}_k$ in the next iterations. Moreover, a step control algorithm is forced upon the suggested step length s_k^{sugg} for the parameter update. Currently, this algorithm is fairly simple. The suggested step lengths for each parameter are treated separately, and only step lengths smaller than a fraction $\alpha_{\max}$ of the current parameter λ_i are accepted.

The essence of the algorithm is as follows: We consider the different parameters separately, and update one parameter at the time. We input the current parameter λ_i and the suggested step for that parameter, denoted by s^{sugg}. The fraction $\alpha_{\max}$ is used to determine the maximum valid step in absolute value, which is the product $s_{\max} = \lambda_i \alpha_{\max}$. The suggested step length is then controlled, and we output the updated parameter, which is given by

$$\lambda_i + \min\left(s_{\max}, \max\left(0, s_{\text{sugg}}\right)\right).$$

The detailed algorithm for one parameter is given below.

```
Input current parameter $\lambda$ and suggested step $s^{\text{sugg}}$.
Choose $0 < \alpha_{\max} < 1$.
Define $s_{\max} \leftarrow \lambda\alpha_{\max}$ as the maximum step length.
if $|s^{\text{sugg}}| > s_{\max}$ then
  if $s^{\text{sugg}} > 0$ then
    return $\lambda + \alpha_{\max}$
  else
    return $\lambda - \alpha_{\max}$
  end if
else
  return $\lambda + s^{\text{sugg}}$
end if
```

Algorithm 5: A step-control algorithm

6.2 L-hypersurfaces

A L-hypersurface parameter choice method generalizes the L-curve method introduced in Section 5.2. For multiparameter optimization, with multiple regularization terms, each of the regularization terms is plotted along an axis. If the number of regularization terms exceeds 2 in Tikhonov regularization, the hypersurface will be (at least) three-dimensional, and thus difficult to visualize.

For total generalized variation, with its two regularization terms, the plotted surface will be two-dimensional. Thus it is still possible to visualize the surface in a meaningful way. Along the first axis, the residual $\|u - v\|_2^2$ is plotted, while the regularization terms are plotted on the two other axis. Similar to what we did for the total variation problem, a logarithmic transformation will be applied on all vectors before plotting them.

Whereas calculating the maximum Gaussian curvature is feasible in two dimensions, larger problems arise in higher dimensions. The calculation of the curvature itself is computationally expensive, and makes line search approaches less attractive. In [BKM02], the authors also point out and show that the curvatures often have many extrema. Therefore, gradient ascent searches are likely to identify local maximizers only, and the problem with computationally expensive curvature calculations remain a problem also for the gradient ascent approach.

A suggested solution to these difficulties is to introduce a **minimum distance function**. This concept was introduced by [BKM02], which applies minimum distance functions and minimizations of these as parameter choice methods both for single- and multiparameter optimization problems. The key elements and the central calculations of the approach will be explained here. For details and all calculations, I refer to chapters 3.1 and 3.2 in [BKM02], where the function first was introduced.

6.2.1 The minimum distance function

Recall that the a general Tikhonov multi-parameter optimization problem with N regularization terms and a linear operator A can be denoted

$$\hat{u} = \arg\min_u \|Au - v\|_2^2 + \sum_{i=1}^{N} \lambda_i \psi_i(u). \tag{6.6}$$

The L-hypersurface is divided into parts of the surface where one of the regularization terms (or the data discrepancy term) dominates the others, and one part where the trade-off between the regularization terms is more balanced. For each regularization term ψ_i, it is possible to choose a coordinate b_i, such that for all $\hat{b_i} \geq b_i$, ψ_i dominates the other terms in the plot. Similarly, we choose a coordinate b_0 where the same domination begins for the data discrepancy term. We can then define the origin O of our L-hypersurface plot as

$$O = (b_0, b_1, \dots b_N)^T.$$

The shortest distance between O and the L-hypersurface itself is given by the length of the vector between the origin and the point on the hyper-surface closest to the origin. We introduce and define a function that can be applied to determine this vector.

Definition 6.1. The **minimum distance function** $v(\boldsymbol{\lambda})$ from the origin O is given as

$$v(\boldsymbol{\lambda}) = \left(\log\Big(\frac{1}{2}\|u_{\boldsymbol{\lambda}} - v\|_2^2\Big) - b_0 \right)^2 + \sum_{i=1}^{N} \left(\log\Big(\psi_i(u_{\boldsymbol{\lambda}})\Big) - b_i \right)^2 \tag{6.7}$$

From here we can continue by defining the **minimum distance point** (MDP) as the point on the L-curve for which the minimum distance function v has a local minimizer $\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}\in\mathbb{R}^N_{(+)}} v(\boldsymbol{\lambda})$. In order to find this optimal vector of regularization parameters $\hat{\boldsymbol{\lambda}}$, a fixed point method is considered. The fixed point method is computationally less expensive than other optimization approaches. It is common to include calculations of partial derivatives in these methods, and that is just the calculations we tried to avoid in the first place!

When we defined the minimum distance function, we used the natural logarithm as the scaling transformation applied before plotting. For the fixed point method we will consider a more general scaling transformation, to ensure that the results are valid for a broader range of transformations than the logarithmic one. We denote the transformation functional by θ. Thus we can express the minimum distance function with a general scaling transformation θ as

$$v(\boldsymbol{\lambda};\theta) = \left(\theta\Big(\frac{1}{2}\|u_{\boldsymbol{\lambda}} - v\|_2^2\Big) - b_0 \right)^2 + \sum_{i=1}^{N} \left(\theta\Big(\psi_i(u_{\boldsymbol{\lambda}})\Big) - b_i \right)^2.$$

We want to find minimizers of this functions, and a natural approach is to identify point in which the gradient is zero. Therefore, we differentiate v with respect to all regularization parameters λ_j for $1 \leq j \leq N$. The partial derivatives of v with respect to the different regularization parameters can be expressed as

$$\begin{aligned}\frac{\partial v}{\partial \lambda_j} =& \Big(\theta(\frac{1}{2}\|u_{\boldsymbol{\lambda}} - v\|_2^2) - b_0\Big)\theta'(\frac{1}{2}\|u_{\boldsymbol{\lambda}} - v\|_2^2)\frac{\partial(\frac{1}{2}\|u_{\boldsymbol{\lambda}} - v\|_2^2)}{\partial \lambda_j} \\ &+ \sum_{i=1}^{N}\Big[\theta\Big(\psi_i(u_{\boldsymbol{\lambda}})\Big) - b_i\Big]\theta'\Big(\psi_i(u_{\boldsymbol{\lambda}})\Big)\frac{\partial(\psi_i(u_{\boldsymbol{\lambda}}))}{\partial \lambda_j},\end{aligned}$$

and whenever the gradient is zero, then all these partial derivatives also are. So the requirement becomes $\frac{\partial v}{\partial \lambda_j} = 0$ for $1 \leq j \leq N$, with the expression for $\frac{\partial v}{\partial \lambda_j}$ given above.

It can now be shown that

$$\frac{\partial v}{\partial \lambda_j} = \sum_{i=1}^{N} -\lambda_i \frac{\partial(\psi_i(u_{\boldsymbol{\lambda}}))}{\partial \lambda_j}.$$

This calculation is omitted from this thesis, but can be found in for example Equations 13-15 in [BKM02]. Note that the authors of that paper uses a different notation from the one introduced above. The result, however, is the same in both cases. Our N equations are simplified, and we have the following system of equations for all j such that $1 \leq j \leq N$:

$$\begin{aligned} \sum_{i=1}^{N} \frac{\partial(\psi_i(u_{\boldsymbol{\lambda}}))}{\partial \lambda_j} \Bigg(& \theta'(\psi_i(u_{\boldsymbol{\lambda}}))\Big(\theta(\psi_i(u_{\boldsymbol{\lambda}})) - b_i\Big) \\ & -\lambda_i \theta'(\frac{1}{2}\|u_{\boldsymbol{\lambda}} - v\|_2^2)\Big(\theta(\frac{1}{2}\|u_{\boldsymbol{\lambda}} - v\|_2^2) - b_0\Big)\Bigg) = 0. \end{aligned} \tag{6.8}$$

From here, we rewrite the system of equations as a matrix-vector system. We note that the parenthesis is independent of j, whereas the partial derivatives depend on both i and j. Using the same notation as in [BKM02], we define

$$\Big[J\Big]_{j,i} = \frac{\partial(\psi_i(u_{\boldsymbol{\lambda}}))}{\partial \lambda_j},$$

and moreover that $\Big[r\Big]_i$ is equal to the big parenthesis in Equation (6.8). We have now described a matrix-vector system $Jr = 0$. From linear algebra we know that this system has only the trivial solution $r = 0$ if J is a non-singular matrix. That is, if J has rank N, then the linear system only has the trivial solution. It can be shown via a geometric argument valid on regular surfaces that J in fact is non-singular[BKM02].

This result means that for $1 \leq i \leq N$, we have the following equality

$$\theta'(\psi_i(u_{\boldsymbol{\lambda}}))\Big(\theta(\psi_i(u_{\boldsymbol{\lambda}})) - b_i\Big) - \lambda_i \theta'(\frac{1}{2}\|u_{\boldsymbol{\lambda}} - v\|_2^2)\Big(\theta(\frac{1}{2}\|u_{\boldsymbol{\lambda}} - v\|_2^2) - b_0\Big) = 0.$$

We can solve this equation for λ_i, and arrive at

$$\lambda_i = \frac{\theta'(\psi_i(u_{\boldsymbol{\lambda}}))}{\theta'(\frac{1}{2}\|u_{\boldsymbol{\lambda}} - v\|_2^2)} \frac{\theta(\psi_i(u_{\boldsymbol{\lambda}})) - b_i}{\theta(\frac{1}{2}\|u_{\boldsymbol{\lambda}} - v\|_2^2) - b_0}. \tag{6.9}$$

This expression can be used as a fixed point iteration for the regularization parameters. We will now consider the special case where $\theta(t) = \log(t)$, which we will apply as the scaling transformation in our numerical implementations. Note that $\theta'(t) = \frac{1}{t}$ in this case. The fixed point iteration can now be written as

$$\lambda_i^{(k+1)} = \frac{\frac{1}{2}(\|u_{\boldsymbol{\lambda}^{(\mathbf{k})}} - v\|_2^2)}{\psi_i(u_{\boldsymbol{\lambda}^{(\mathbf{k})}})} \frac{\log\Big(\psi_i(u_{\boldsymbol{\lambda}^{(\mathbf{k})}})\Big) - b_i}{\log(\frac{1}{2}\|u_{\boldsymbol{\lambda}^{(\mathbf{k})}} - v\|_2^2) - b_0}. \tag{6.10}$$

7 Numerical experiments with multi-parameter optimization

We will now turn out attention towards performing numerical experiments with the two multi-parameter choice method that were introduced in the last section. Unless otherwise is explicitly stated, we use the noise level $\sigma = 0.1$ throughout this section, and use the denoising algorithm for total generalized variation as the primal-dual solver. All experiments are performed channel-wise, and the pixel intensities are scaled down to the unit interval as before. We will investigate Broyden's algorithm for the balanced discrepancy principle, as well as the fixed point iteration based on the minimum distance function on a L-hypersurface. The algorithms and their resulting reconstructions are compared with the best results obtained by the existing single-parameter methods tested in Section 5.

The numerical experiments have mainly been concerned with searching for the optimal vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)^T \in \mathbb{R}^2_{(+)}$ that contains the two regularization parameters for total generalized variation. The PSNR and SSIM metrics have been used as the objective comparison tool, which measure how close a reconstructed image is to the original, noise-free image.

A natural starting point may be to ask whether it is at all feasible to obtain better denoising results by applying total generalized variation than total variation. The initial experiment that was performed to test this has the following structure. Consider a grid of possible parameter choices for λ_1 and λ_2. For each possible pair of parameters, we apply the TGV denoising algorithm on a noisy image that was generated from the KODAK image (Figure 2.1) with noise level 0.1. We then evaluate the reconstruction in the PSNR-matrix, and plot the results versus the parameter choices.

The results from the experiments are shown in Figure 7.1. We can see that the highest PSNR scores exceed 28.20, which was the optimal value for total variation denoising at that noise level. Thus, we have shown through our numerical experiments that it is feasible to denoise by TGV, and to perform better than the optimal reconstructions using TV. Note that this brute force-approach is in no way optimalized, it simply tests 525 different sets of parameters. The takeaway from this experiment is that we, by applying TGV denoising, are able to produce better results than we were with TV. It is not essential at this stage to focus on which parameter choices have led to the best reconstructions. However, the level curves in Figure 7.1 give indications on good initial guesses for our parameter selection algorithms.

Now that we know that good parameter choices exist, we will turn to the multi-parameter choice methods that were introduced in the last section, in order to investigate whether they produce better reconstructions in the PSNR metric than the single-parameter choice methods that have been implemented and described in Section 5.

7.1 Discrepancy principle with Broyden's method

The first method that we have applied for multi-parameter optimization is the discrepancy principle with Broyden's algorithm for total generalized variation with a fixed noise level $\sigma = 0.1$. The method and the exact algorithm was introduced in Section 6.1.1. Recall that this method was based on the balanced discrepancy principle. That is, the noise level is known to the algorithm, and we try to balance

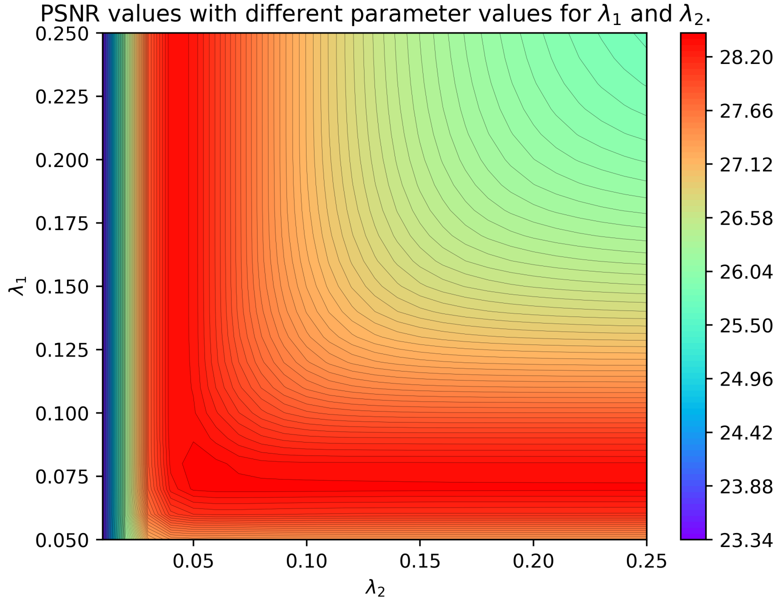


Figure 7.1: PSNR scores of reconstructions with 525 different parameter sets (λ_1, λ_2)

.

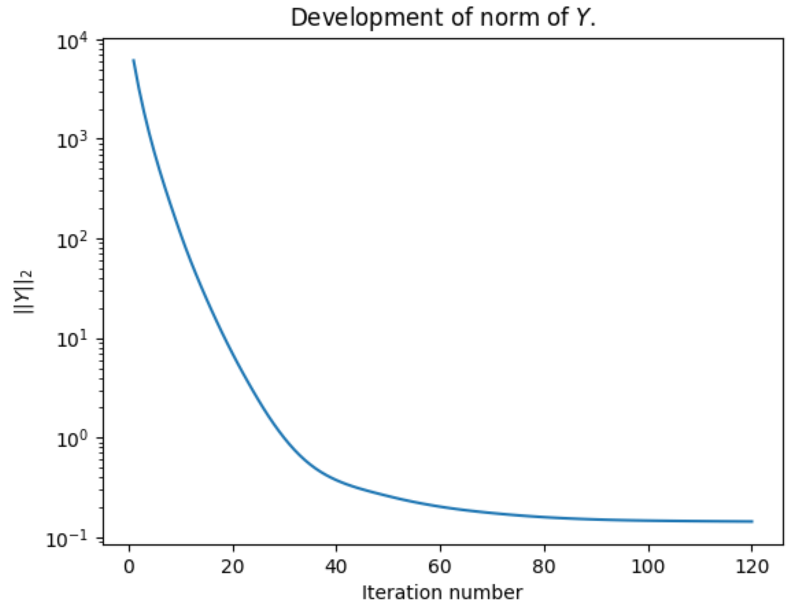
the influence of two regularization terms. Broyden's method is used to find the zeros of the function $\mathbf{Y}$, that is given in Equation (6.4). Recall that this function was given as

$$\mathbf{Y} = \begin{bmatrix} \frac{1}{2}\|u^\sigma_{\boldsymbol{\lambda}} - v\|_2^2 - \frac{1}{2}\sigma^2 + \lambda_2\psi_2(\mathbf{w}) - \lambda_1\psi_1(u^\sigma_{\boldsymbol{\lambda}}, \mathbf{w}) \\ \frac{1}{2}\|u^\sigma_{\boldsymbol{\lambda}} - v\|_2^2 - \frac{1}{2}\sigma^2 + \lambda_1\psi_1(u^\sigma_{\boldsymbol{\lambda}}, \mathbf{w}) - \lambda_2\psi_2(\mathbf{w}) \end{bmatrix}.$$

The aim of the numerical approach is to find the $u^\sigma_{\boldsymbol{\lambda}}$ that minimizes the Euclidean norm $\|\mathbf{Y}\|_2$ of the function. Algorithm 4 is applied to the initial noisy data with some initial guess $\boldsymbol{\lambda_0}$ for the regularization parameters. Based on the results in from the brute force-simulation presented in Figure 7.1, we choose $\boldsymbol{\lambda_0} = [0.07, 0.07]^T$ as our initial guess. Moreover, we choose to perform $K = 120$ steps. As we will see, this is enough to ensure convergence of the algorithm, and that the PSNR measure of the reconstructed image reaches an acceptable level. It is worth mentioning that we still perform the denoising separately in each of the three colour channels.

In order to illustrate the progress of Broyden's method as iterations are performed, we plot the development of the Euclidean norm of the residual $\mathbf{Y}$. From Figure 7.2, we can see that Broyden's method is successful in minimizing the norm of the function $\mathbf{Y}$. We present the development of the absolute value of the norm, and can note that the relative norm is improved almost by a factor of 10^5. The initial norm $\|\mathbf{Y}_0\|_2$ is close to 10^4, whereas the norm of the vector converges to approximately 10^{-1} as the iterations are run.

Figure 7.2: The norm of the residual $\mathbf{Y}$.



So the method is successful in finding the zeros of the function, but what about the reconstructions and their PSNR-scores? The reconstruction itself is presented in Figure 7.3.

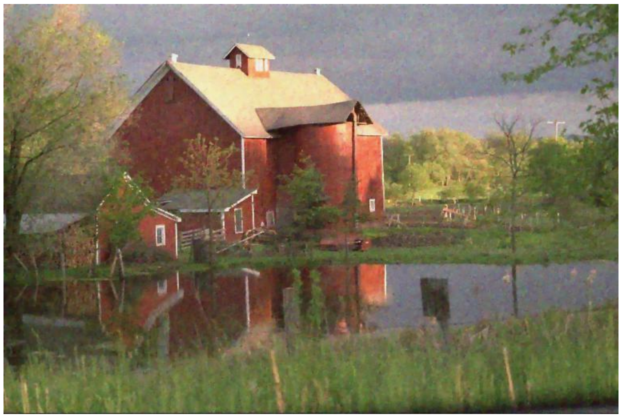


Figure 7.3: The reconstructed image after 120 iterations of Broyden's method.

The reconstructed image has a PSNR score of 28.28, which is higher than the score for the single-parameter methods that have been tested earlier. Thus, the multi-parameter TGV approach performs better than total variation approaches, at least when Broyden's method is used as the numerical solver for the system of equations arising from the balanced discrepancy principle. The results and the PSNR score also indicate that the multi-parameter optimization produces better results than the single-parameter TGV approach. We can also consider a plot of how the PSNR score developed through the 120 iterations of Broyden's method. The plot is shown in Figure 7.4, and shows that the method needs relatively few iterations to produce acceptable and good reconstructions. However, the PSNR scores continue to increase with the number of iterations, before it stabilizes at 28.28.

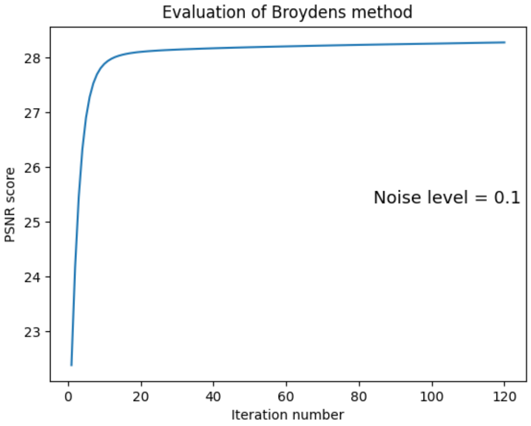


Figure 7.4: The development of PSNR scores using Braydon's method.

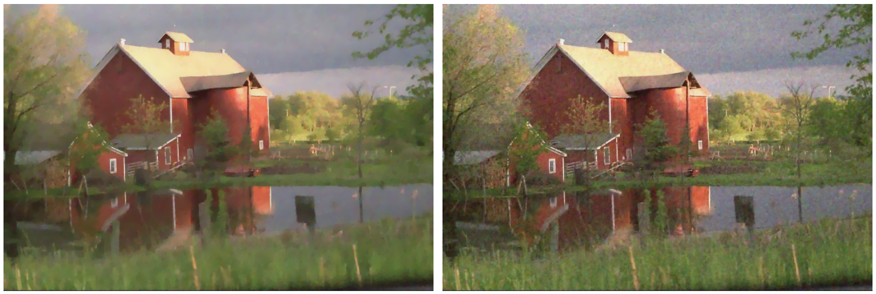
7.2 Comparison between reconstructions with TV and TGV

We have previously adressed the difficulties of establishing an objective measure for the quality of image reconstructions. Both PSNR and SSIM have their strengths and weaknesses, but another factor is also important to consider when evaluating images; their visual impressions. Many images will be observed and interpreted by humans, who tend to have their own subjective means of what a good reconstruction is. Therefore, we will look into differences between reconstructions created by total variation and reconstructions created by total generalized variation. Can we discover and establish some patterns that give insight into the visual differences between the reconstruction techniques?

We consider first the denoised images at noise level 0.1, using the discrepancy principle. In Figure 7.5, we present these images. The PSNR scores of these reconstructions, compared to the original image, is 28.02 for total variation and 28.28 for total generalized variation. According to the objective PSNR measure, the TGV reconstruction should be 'better' than the TV reconstruction.

In order to visualize the difference between these reconstructions, there are several feasible approaches. In Figures 7.6 and the differences in absolute value between u_{TV} and u_{TGV} in the first RGB-channel are presented. The differences are plotted pixelwise. This plot is very similar also for the other RGB-channels, so

Figure 7.5: Reconstructions:



(a) Total variation reconstruction. PSNR = 28.02

(b) Total generalized variation reconstruction. PSNR = 28.28

we will base the following discussion about the differences between the denoising methods on Figure 7.6.

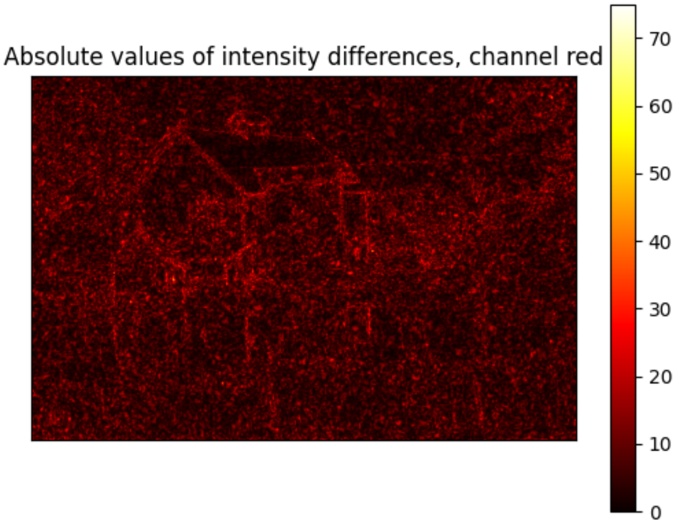


Figure 7.6: Intensity differences in the first RGB channel.

The perhaps most interesting observation from the plot is related to the roof of the barn. We can see that there are small differences on the roof itself, which is almost piecewise constant in intensity. On the ridge of the roof however, which corresponds to edges in the image, we see larger differences between the reconstructions. This is where we have spikes in the pixel intensities of the original image, and it is not surprising to see that different reconstructions are more different on these transitions.

Figure 7.7 shows the absolute value of the intensity differences in the first RGB-channel again, but zoomed in at the roof. From this plot it is easier to see that the differences between the reconstructions are larger on the ridge than on the roof itself. This is another illustration of how edges are treated differently in the two denoising methods.

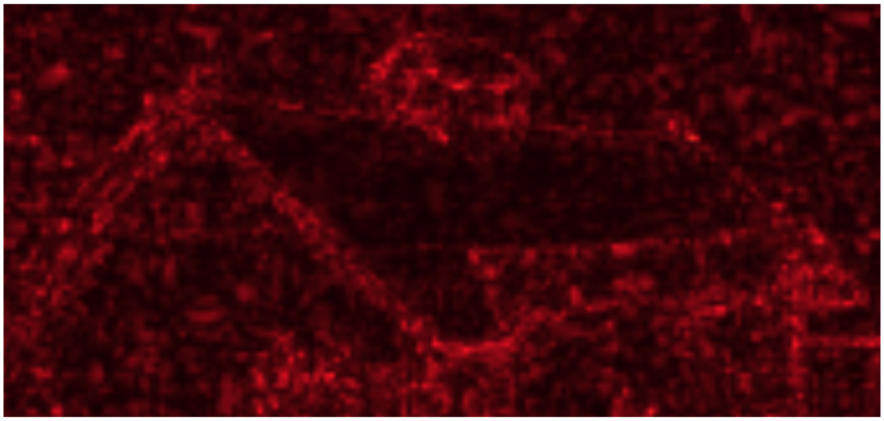


Figure 7.7: Intensity differences in the first RGB channel, zoomed in at the roof

We have however not been able to identify systematic structural differences between the two methods for the detailed barn image. We have only been able to state that the differences are larger close to edges.

7.3 Numerical experiments based on L-hypersurfaces

Obtaining good image reconstructions by applying L-hypersurface based approaches has turned out to be more challenging. We recall from the single-parameter experiments that the L-curve method performed drastically worse than both quadratic regularization and the discrepancy principle for total variation. These problems are just increasing in size when the number of regularization parameters also increases. As before, the method has no knowledge of the expected noise level σ, and is thus a purely data-driven problem.

In our numerical experiment we start with a noisy version of the barn image, with noise level $\sigma = 0.1$. The denoising is, as before, applied channelwise. We then create a three-dimensional L-hypersurface by testing different regularization parameters, and for each set of parameters we store the residual and the norm of each of the two regularization terms in total generalized variation. After performing a logarithmic transformation of all these three `Numpy` arrays, we make a three-dimensional plot of the resulting arrays. The appearance of this plot depends on which regularization parameters that are tested. In Figure 7.8, a L-hypersurface based on 225 different parameter sets, in one colour channel, is presented. The 225 parameters sets that have been used to create the L-hypersurface are all possible combinations of $\boldsymbol{\lambda} = (0.01i, 0.01j)$ for $1 \leq i \leq 15$ and $1 \leq j \leq 15$. The idea in the numerical algorithm is then to use the fixed point iteration that was derived in Equation (6.10), in order to update $\boldsymbol{\lambda}$ and identify the optimal regularization parameters for this denoising problem.

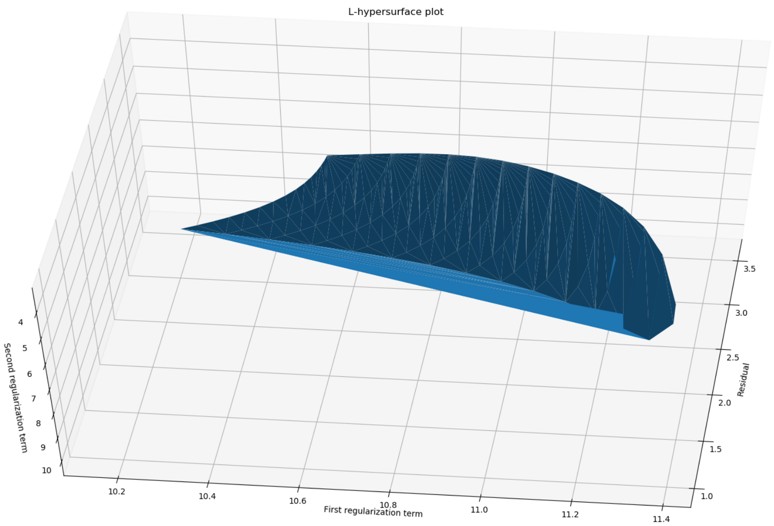


Figure 7.8: The geometry of the L-hypersurface.

As we can see from Figure 7.8, it is challenging to exactly determine the origin of the L-hypersurface, that ought to be used as the origin in the minimum distance function (6.7). This origin is needed for the fixed point iteration, as the algorithm want to find the set of regularization parameters that minimizes the distance from the hypersurface to this origin. A three-dimensional scatter plot of the 225 different combinations of the regularization parameters strengthens the intuition of the geometry of the hypersurface, and is presented in Figure 7.9. The effect of the discretization is also visible in this figure, where the different colours represent the value of the product $\lambda_1\lambda_2$.

From Figure 7.9 it is easier to see that the three-dimensional L-hypersurface has similar problems as the two-dimensional L-curve, and that the curvature "bends" in the opposite direction of what we hoped for. Another problem is that we only have a visible curvature in one direction. The form of the L-hypersurface is very valley-like. This is likely to cause numerical unstabilities and also cause non-uniqueness of solutions that lie within this 'valley'.

We have tested two possible approaches to determine the origin based on Figure 7.9. We have chosen the point r_1 on the discretized L-hypersurface that has the minimum combined distance to the 224 other points. We have also chosen the point r_2 that has the minimum distance to all the 225 points, without requiring r_2 to be on the surface itself. r_1 and r_2 has then been used as origins in the fixed point iteration (6.10). Both pre-selected and random starting points have been tested, and do not seem to affect the results of the fixed point iteration. For the experiments that have been included in this text, the initial vector of parameters $\boldsymbol{\lambda}_0$ have been created by drawing two random numbers between 0.05 and 0.15 from

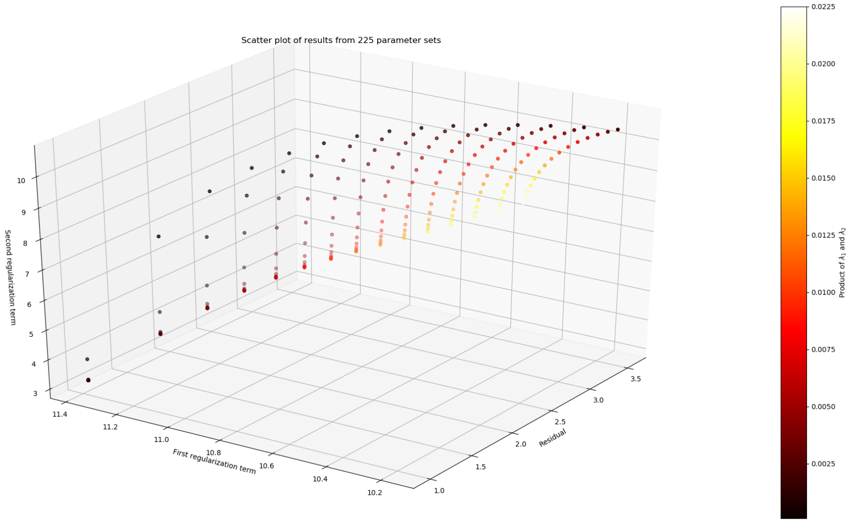


Figure 7.9: The L-hypersurface as a scatter plot.

a uniform and continuous distribution.

The results of the fixed point iteration are presented as three PSNR image comparisons. We compare the reconstruction u_L iteration by iteration to the original image and to the reconstruction u_{TGV} that was produced by the balanced discrepancy principle and Broyden's method. Moreover, we add the PSNR score of u_{TGV} and the original image as a tool of comparison. These results and the development of the quality of the reconstructions are presented in Figure 7.10 with origin r_1 and Figure 7.11 with origin r_2.

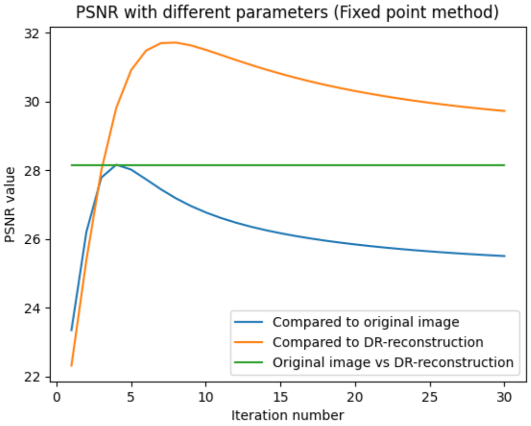


Figure 7.10: PSNR scores after the first 30 iterations with origin r_1.

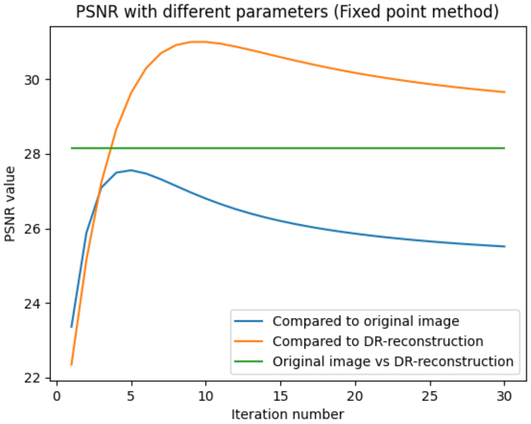


Figure 7.11: PSNR scores after the first 30 iterations with origin r_2.

In both cases, the fixed point iteration converges to a parameter choice that produces reconstructions with PSNR scores of 25.5. We can notice that with origin r_1 we do at one stage have a reconstruction of about the same quality as u_{TGV}. The fixed point iteration is however far from converged at this stage, and the algorithm proceeds to choosing worse parameters, without ”discovering” how close it was, since the original image is unknown to the denoising algorithm.

The reconstructed image obtained by using the origin r_1 can be seen in Figure 7.12. We can see clear indications of over-regularization, similar to those we observed in the cases with too high regularization parameters for total variation and quadratic regularization.

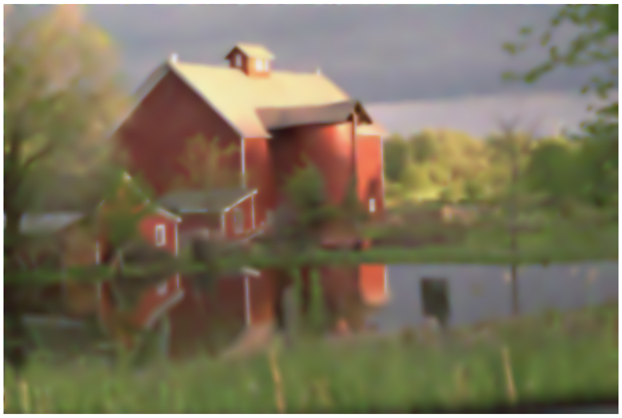


Figure 7.12: Reconstructed image by fixed point iteration and use of L-hypersurfaces.

The results are thus similar to the results for the two-dimensional L-curve.

The L-hypersurface method produces worse reconstructions than the (balanced) discrepancy principle. Again, this is an expected result as the discrepancy principle has more knowledge (specifically, the noise level) about the problem. The main conclusion from this subsection, however, is that the L-hypersurface method is not a well-suited method for image reconstructions with total generalized variation. The geometry of the hypersurface is not easy to cope with when determining curvatures or corners, with non-uniqueness of solutions as the most challenging problem.

7.4 A comparison of the best reconstruction methods

From our analysis until this stage, we have concluded that three of the parameter choice methods we have tested are able to produce good reconstructions to images. These three methods are the discrepancy principle for total variation, the discrepancy principle for single-parameter TGV denoising, and finally a multi-parameter method, namely the balanced discrepancy principle with Broyden's method, which also is using TGV denoising. In this subsection we will take a closer look at how these three methods compare to eachother for the standard noise level $\sigma = 0.1$, in terms of both PSNR scores, runtimes and visual impressions.

For the numerical experiment, we reuse the previously derived and explained algorithms. For the discrepancy principle, we use Algorithm 3. As before, we use $q = 0.9$ and $t_0 = 0.5$, and choose the first reconstruction u_{TV} for which $\|u_{\mathrm{TV}} - v\| < \sqrt{nm}\sigma$, where v is the noisy image with noise level σ. We use the same algorithm for the single-parameter approach with TGV, and we still use $\gamma = \frac{1}{2}$ as the fixed relation between the parameters λ_1 and λ_2.

For the balanced discrepancy principle with use of Broyden's method we combine the algorithm itself (Algorithm 4) with the step-control algorithm in Algorithm 5. We use $\alpha_{\max} = 0.05$ as the maximum relative stepsize. For the initial vector of parameters $\boldsymbol{\lambda}_0$, we use $[0.07, 0.07]^T$. Tests with more randomized initial values seem to converge to some local minima, so this vector is forced as the value for $\boldsymbol{\lambda}_0$. We use a relative convergence criteria for the norm of the residual in Broyden's method, and stop the iterates when this relative norm is reduced to $0.3 \cdot 10^{-4}$ or less.

7.4.1 Introducing two new images

All experiments until this point have been performed on the barn images. We now want to include two more images with different properties. The KODAK data set remains the source of images, and we have chosen KODAK images number 9 and 13 as the new images[Fra10a; Fra10b]. Image number 9 is presented in Figure 7.13. This shows some sailboats at the ocean, and has relatively few details. The sail itself is mainly white, whereas the ocean is blue. The somewhat clouded sky is also a large part of the image. This image will be referred to as 'the sailboat image' in the following. The sailboat image has the same number of pixels as the barn image, but the dimensions are now reversed, being 768 times 512.

The other new image, KODAK image number 13, will be referred to as 'the river image'. This image has many details and is a complicated image to reconstruct. Both small details as rocks and streams in the river, the trees and bushes behind the water and the mountain makes this image a more complicated one than the

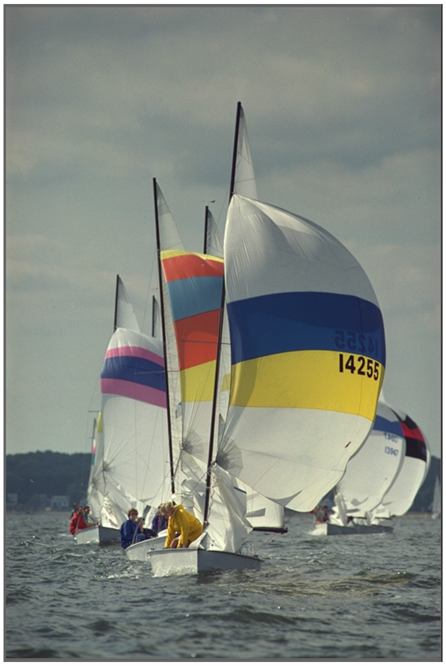


Figure 7.13: KODAK image number 9, sailboats on the ocean. Photo by: John Menihan.

ones previously introduced. The image has the same dimensions as the barn image, with 562 times 768 pixels. The river image is seen in Figure 7.14.

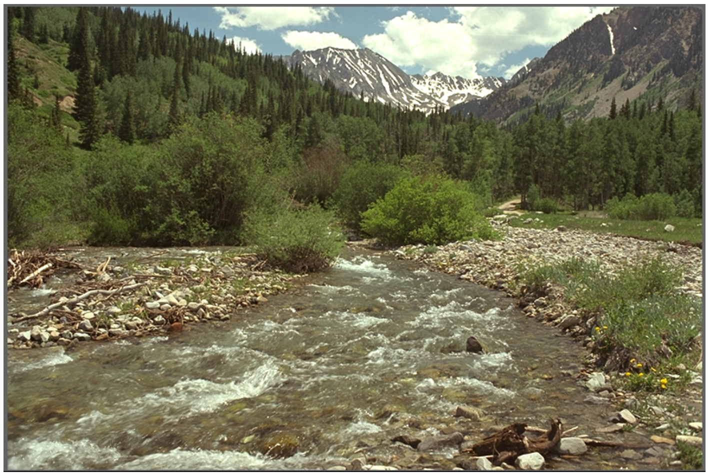


Figure 7.14: KODAK image number 13, the river image. Photo by: Norm Kerr.

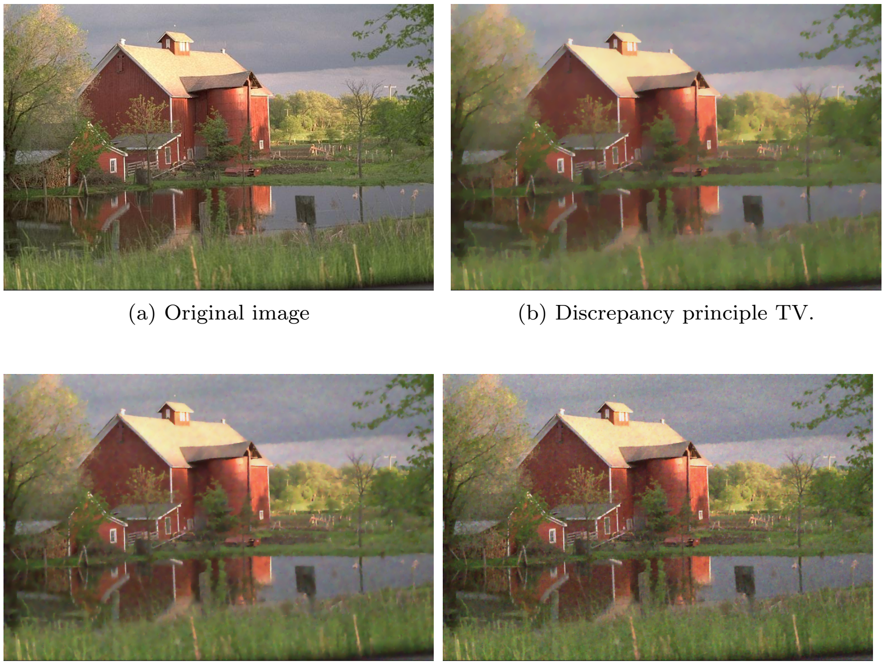
7.4.2 Results for the barn image

Method	PSNR score	Runtime
TV DP	28.02	107 s
Single-parameter TGV with DP	28.23	98 s
Broyden's method for balanced DP with TGV	**28.28**	533 s

Table 7.1: Results for the barn image.

The results for the barn image have been discussed to some extent earlier in the text, in terms of quality of the reconstructions. The full results are presented in Table 7.1, and the only new relevant information is the runtime of the different algorithms. We notice that the single-parameter choice algorithms use less than two minutes to choose a parameter and to reconstruct the image. Broyden's method for the balanced discrepancy principle uses up to eight minutes, but produces a better reconstruction. It is worth noting that Broyden's algorithm needs this extra time to improve an already good reconstruction. As we may recall from Figure 7.4, Broyden's method does not use many iterations to obtain a good reconstruction. But slowly, and using some time, the reconstruction is improved to a level where it becomes a superior reconstruction compared to those created by the other methods.

Figure 7.15: Original and reconstructed barn images.



(a) Original image

(b) Discrepancy principle TV.

(c) Single-parameter TGV with discrepancy principle.

(d) Balanced discrepancy principle with Broyden's method.

7.4.3 Results for the river image

Method	PSNR score	Runtime
TV DP	23.87	117 s
Single-parameter TGV with DP	23.92	104 s
Broyden's method for balanced DP with TGV	**24.40**	435 s

Table 7.2: Results for river image.

The results for different methods for the river image are presented in Table 7.2. The immediate observation is that Broyden's method for the balanced discrepancy principle is superior to the two others also for this image. Moreover, we observe that the PSNR scores are lower than for the barn image. The images are of the same size, so it is reasonable to believe that the difference is caused by the more complex details of the river image, which are more difficult to reconstruct. The runtimes are similar for this image. Again, Broyden's method is able to produce a reasonable reconstruction within the same time period as the two other methods, but when it runs for between seven and eight minutes it produces a superior reconstruction.

The reconstructed images themselves are presented in Figure 7.16. It is especially worth noting that the total variation reconstruction (Figure 7.16b) again shows indications of being blurry and unclear, in particular in the bushes. In particular the best reconstruction, created by Broyden's method and seen in Figure 7.16d, contains more details and visible differences between the pixels.

7.4.4 Results for the sailboat image

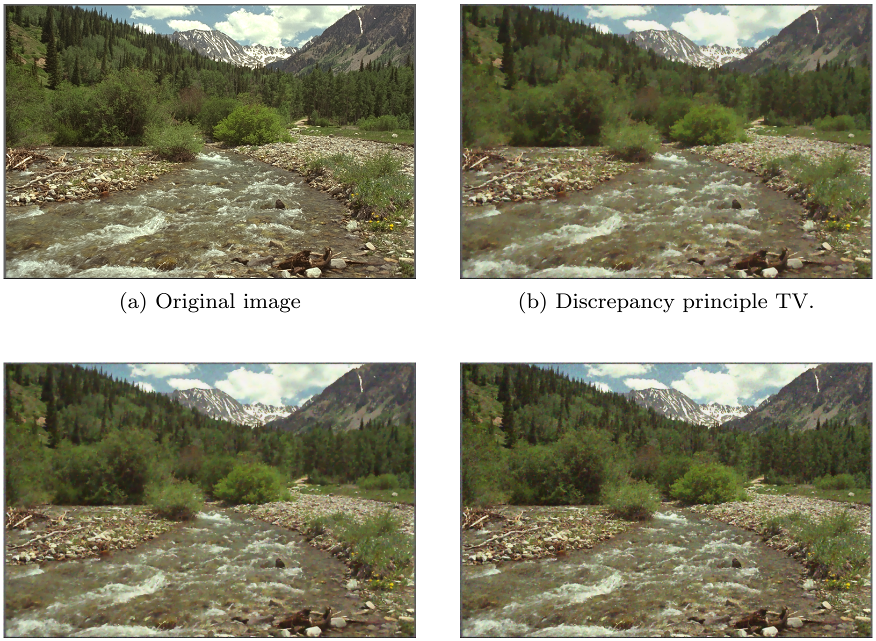
Method	PSNR score	Runtime
TV DP	**29.42**	109 s
Single-parameter TGV with DP	29.34	91 s
Broyden's method for balanced DP with TGV	29.19	442 s

Table 7.3: Results for sailboat image.

The results for the different parameter choice methods for the sailboat image are presented in Table 7.3. These results are immediately interesting, as they challenge our current understanding of which algorithms that are best suited for image denoising. We can see that the total variation algorithm has the highest PSNR score, and a visual inspection of the reconstructions strengthen that impression. The reconstructions can be seen in Figure 7.17. This is an image for which where the staircasing property of total variation turns out to be advantageous. Both the TGV reconstruction approaches fail to remove noise to the extent that the TV-approach does. The image contains few areas with almost constant intensity of different colours, which is the perfect situation for a total variation approach that often produces staircased results [Hub+19].

The visual impression from the reconstructed images is that the sky is reconstructed and denoised best in the total variation reconstruction (Figure 7.17b), while the ocean is a bit too smooth in that reconstruction. The total variation method fails to accurately reconstruct the details of the waves and colour nuances

Figure 7.16: Original and reconstructed river images.



(a) Original image

(b) Discrepancy principle TV.

(c) Single-parameter TGV with discrepancy principle.

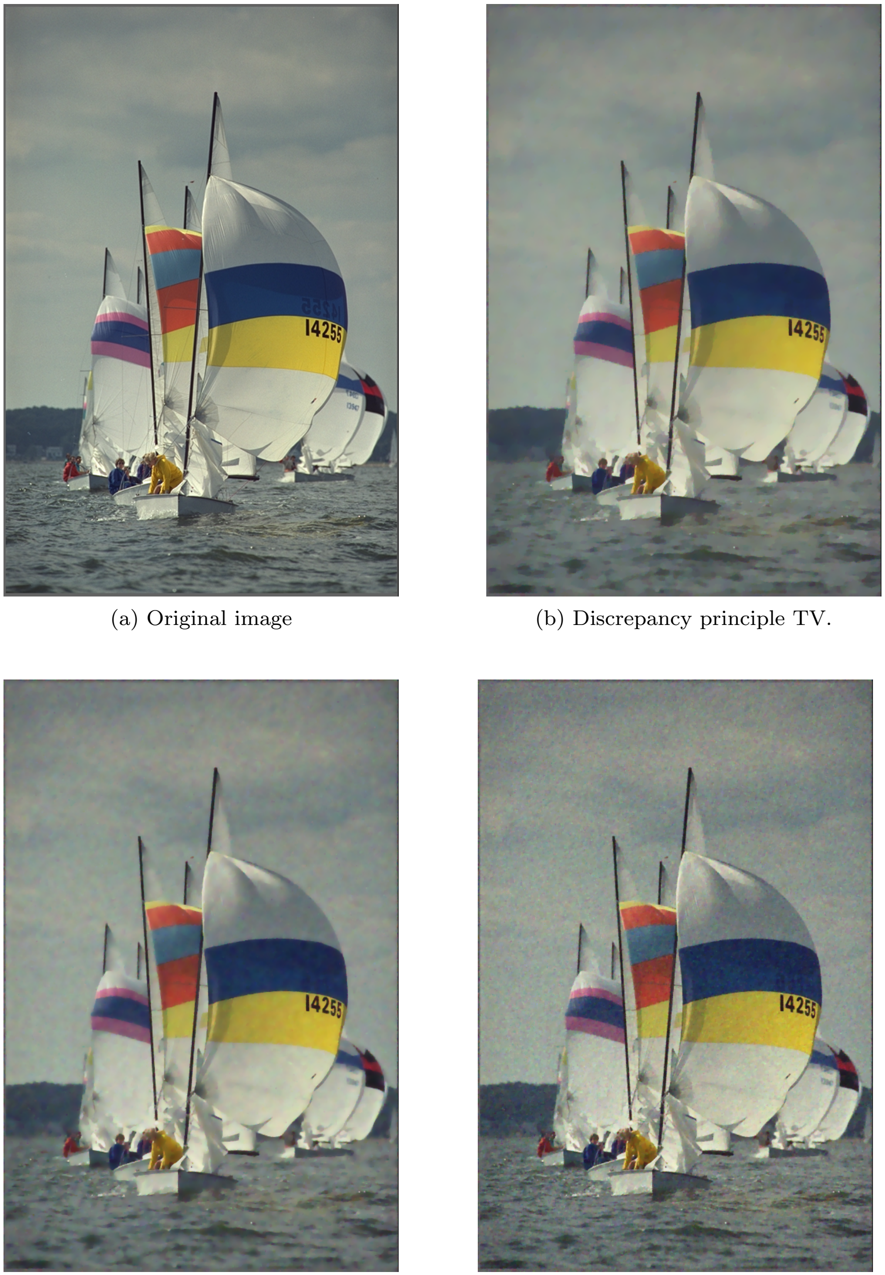
(d) Balanced discrepancy principle with Broyden's method.

in the water. When only considering the ocean, the Broyden's method reconstruction again contains the most details, as it did for the water in the river image. It is also an interesting observation that the people in the boat are much clearer in (in particular) the Broyden reconstruction than the total variation reconstruction. This points to the fact that total variation is a well-suited method for larger surfaces and images with close to piecewise constant pixel intensities, whereas the method has major shortcomings for images with more details and edges.

These differences touch the most relevant differences between total variation denoising and the TGV-based approaches. We can zoom in on the images to make the differences even more visible and easier interpretable. In Figure 7.18, the bottom part of the sailboat image is shown, for the total variation reconstruction and the TGV-based Broyden reconstruction respectively. The images are clearly different, and the TGV reconstruction contain mnay more details and also show a more realistic transition between the water and the sailboat.

If we also zoom in at the sky and ocean, we see the strengths of total variation more closely. This is done in Figure 7.19, where we can see that the piecewise constant approach pixel intensities on the sail and in the background leave a better visual impression than the more noisy TGV reconstruction. The staircasing effect, which we normally consider a disadvantage for denoising purposes, becomes an advantage as the original image mainly consist of areas with piecewise constant intensities. The denoising results for this image is particularly interesting, as they illustrate the strengths and weaknesses of the different regularization functionals.

Figure 7.17: Original and reconstructed sailboat images.



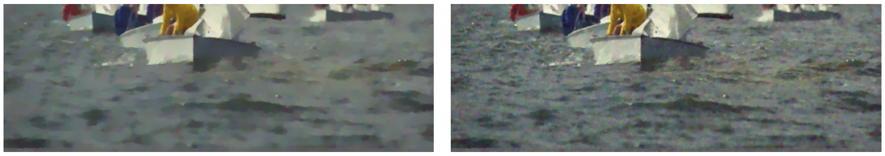
(a) Original image

(b) Discrepancy principle TV.

(c) Single-parameter TGV with discrepancy principle.

(d) Balanced discrepancy principle with Broyden's method.

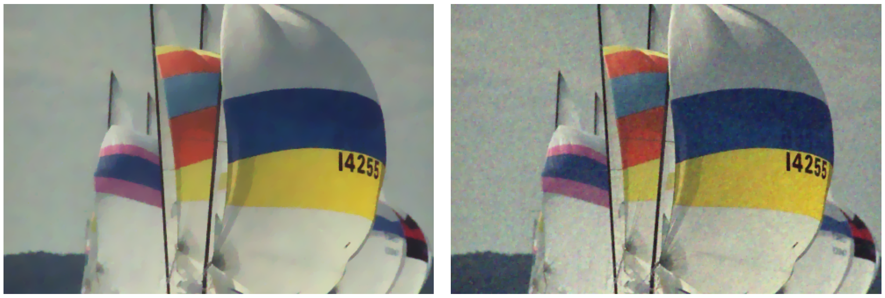
Figure 7.18: Zoom at the ocean in the reconstructions.



(a) Total variation.

(b) TGV.

Figure 7.19: Zoom at the sails and sky in the reconstructions.



(a) Total variation.

(b) TGV.

8 Conclusions

The two numerical multi-parameter methods we have tested tell quite different stories. The L-hypersurface method experiences many of the same challenges as the two-dimensional L-curve, also in relation to the topology of the curve. It is challenging both to determine an accurate origin for the multi-dimensional fixed point iteration, and to decide on an objective criteria for how to choose this origin. The fixed point iteration itself does converge to a choice of regularization parameters that restore some of the lost information in the signal. However, the reconstructed signal is more noisy than the reconstructions originated from the optimal choices of regularization parameters.

Broyden's algorithm for the balanced discrepancy principle method, on the other hand, turns out to be very effective in denoising images at low noise levels. In about 100 iterations, depending on the properties of the relevant image, the norm of the residual is reduced by a factor of almost 10^5. The images that are reconstructed using this method are in general more similar to the original and noise-free images than the reconstructions that are generated using the existing single-parameter methods, at least in the situations where the images contains many edges and details.

Single-parameter total variation remains the best method for images with few details and many areas with close to constant pixel intensities, such as the sailboat image. The discrepancy principle is an efficient parameter choice method in this situation. Total variation is capable of producing better reconstructions of images than quadratic regularization, given good parameter choices. An automated parameter choice implementation of the discrepancy principle, for a known noise level, performs better than both the optimized quadratic regularization and the L-curve method. For high noise levels, the discrepancy principle made near optimal parameter choices for total variation.

The best numerical results, both for one and two regularization parameters, have been produced by applying different approaches based on the discrepancy principle. The challenge in a more general denoising process is that we can not always expect the (expected) noise level to be known. We have not been able to identify equally efficient parameter choice methods for situations where the expected noise level is unknown.

All tests have been performed on RGB-images with 393216 pixels. The runtimes of the methods are not worrying. For the single-parameter approach, the discrepancy principle-based algorithm makes its parameter choice in less then two minutes. For Broyden's method in the multi-parameter setting, the algorithm runs for seven to eight minutes before reaching convergence.

Appendices

A The conjugate gradient (CG) algorithm

The conjugate gradient algorithm is an iterative algorithm which is used to linear equation systems. The linear equation system can be written as $Au = v$, where we require the (known) matrix A to be positive definite [HS52]. The purpose of the iterations is to find an u such that the system of equations is solved exactly. We denote by u_k the solution after k iterations with the conjugate gradient algorithm.

In the algorithm, it is usual to define a tolerance `TOL` that is used as a convergence criteria. We consider the relative residual of the solution, compared to the residual for the initial guess. More specific, we terminate the algorithm for the first k which satisfies $\frac{\|r_k\|}{\|r_0\|} = \frac{\|Au_k - v\|}{\|Au_0 - v\|} < \texttt{TOL}$.

We start by defining an initial residual $r_0 = \|Au_0 - v\|$ from the initial guess u_0 for u. From here, the algorithm takes the following form [HS52]:

```
Choose a tolerance TOL and let $k = 0$.
Make an initial guess $u_0$.
$r_0 \leftarrow Au_0$
$p_0 \leftarrow r_0$
Calculate $\|r_0\|_2$.
while $\frac{\|r_k\|_2}{\|r_0\|_2} \geq \texttt{TOL}$ do
    $k \leftarrow k + 1$
    $\alpha_k \leftarrow \frac{\|r_{k-1}\|_2}{\langle p_{k-1}, Ap_{k-1} \rangle}$
    $u_k \leftarrow u_{k-1} + p_{k-1}$
    $r_k \leftarrow r_{k-1} - \alpha_k Ap_{k-1}$
    $\beta_k \leftarrow \frac{\|r_k\|_2}{\|r_{k-1}\|_2}$
    $p_k \leftarrow r_k + \beta p_{k-1}$
end while
return $u_k$
```

Algorithm 6: Conjugate gradient algorithm

B An intuition of the TGV functional in a one-dimensional setting

It may be challenging to get a grip of what the total generalized variation algorithm actually does and is. The aim of this part is to strengthen the readers intuition for total generalized variation, by considering a one-dimensional situation.

We will consider a TGV-functional for which $\lambda_1 = \lambda_2 = 1$, meaning that we can remove the parameters from the equation. Moreover, we assume $u, v, w : \mathbb{R} \to \mathbb{R}$. That is, all functions takes real numbers as inputs, and also return real numbers.

v plays the role of the noisy measurement, from which we want to determine a reconstructed solution u. The function w is an approximation to the gradient of u. In one dimension, however, gradients can be called derivatives, and we will use Newtons notation to denote the derivatives. Moreover, the symmetrized gradient of w is in one dimension simply the derivative of w. Thus the minimization problem can we written as

$$T(v) = \min_u \int_{\mathbb{R}} \Big[(u - v)^2 + |u' - w| + |w'|\Big] \mathrm{d}x. \tag{B.1}$$

Let $W' = w$, and then rewrite u as $W + (u - W)$, to obtain

$$T(v) \min_{u,W} \int_{\mathbb{R}} \Big[(W + u - W - v)^2 + |(u' - W')| + |W''|\Big] \mathrm{d}x. \tag{B.2}$$

Now we introduce another variable Z. We let $Z = u - W$. That is, Z is the reconstructed signal minus the anti-derivative of w. The idea is now to observe that u can be split into to parts:

$$u = W + Z.$$

Both the first and the second parenthesis in Equation (B.2) contain a term $u - W$ or $(u - W)'$. Now these terms can be replaced by Z and Z' respectively, to obtain a new minimization problem without u:

$$\min_{W,Z} \int_{\Omega} (W + Z - f)^2 + |Z'| + |W''| \mathrm{d}x. \tag{B.3}$$

We see that we have split u into two parts. One part for which we want to minimize the first derivative, and one part for which we want to minimize the second derivative.

For image denoising we can consider Z as the parts of the image where there are few changes between pixels. This is the area where we are most likely expect constant reconstructions due to staircasing when we apply total variation. In areas where the image does have transitions or spikes however, it does not make too much sense to minimize the gradient, since the original, noise-free image probably has quite a large gradient here. Instead we try to minimize the second derivative in these areas, such that the gradient itself does not vary to much, and we get a relatively smooth gradient on the edges of the image.

References

[And+13] Stefano Andriani et al. “Beyond the Kodak image set: A new reference set of color image sequences”. In: Sept. 2013, pp. 2289–2293. ISBN: 978-1-4799-2341-0. DOI: 10.1109/ICIP.2013.6738472.

[AH56] Kenneth J. Arrow and Leonid Hurwicz. “Reduction of Constrained Maxima to Saddle-point Problems.” In: *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability* 5 (1956), pp. 1–20. DOI: https://projecteuclid.org/euclid.bsmsp/1200511853.

[AK07] Mehiddin Al-Baali and H. Khalfan. “An Overview of Some Practical Quasi-Newton Methods for Unconstrained Optimization”. In: *SQU Journal for Scientific Research: Science and Technology, Oman* 12 (June 2007), pp. 199–209. DOI: 10.24200/squjs.vol12iss2pp199-209.

[BKM02] Murat Belge, Misha E. Kilmer, and Eric L. Miller. “Efficient determination of multiple regularization parameters in a generalized L-curve framework”. In: *Inverse Problems* 18.4 (2002), pp. 1161–1183. ISSN: 0266-5611. DOI: 10.1088/0266-5611/18/4/314. URL: https://doi.org/10.1088/0266-5611/18/4/314.

[BM12] Gilles. Blanchard and P. Mathé. “Discrepancy principle for statistical inverse problems with application to conjugate gradient iteration”. In: *Inverse Problems* 28.11 (2012), pp. 115011, 23. ISSN: 0266-5611. DOI: 10.1088/0266-5611/28/11/115011. URL: https://doi.org/10.1088/0266-5611/28/11/115011.

[Bor17] David. Borthwick. *Introduction to Partial Differential Equations.* Universitext. Springer International Publishing, 2017. ISBN: 9783319489360. URL: https://books.google.no/books?id=KpjlDQAAQBAJ.

[BL06] Jonathan Borwein and Adrian Lewis. *Convex Analysis and Nonlinear Optimization.* 2006, p. 66. DOI: https://doi.org/10.1007/978-0-387-31256-9.

[BKP10] Kristian Bredies, Karl Kunisch, and Thomas. Pock. “Total Generalized Variation”. In: *SIAM J. Imaging Sciences* 3 (Jan. 2010), pp. 492–526. DOI: 10.1137/090769521.

[Bro65] Charles George Broyden. “A class of methods for solving nonlinear simultaneous equations”. In: *Math. Comp.* 19 (1965), pp. 577–593. ISSN: 0025-5718. DOI: 10.2307/2003941. URL: https://doi.org/10.2307/2003941.

[CP11] Antonin Chambolle and Thomas Pock. “A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging.” In: *Journal of Mathematical Imaging and Vision* 40 (2011), pp. 120–145.

[CV20] Christian Clason and Tuomo Valkonen. *Introduction to Nonsmooth Analysis and Optimization.* 2020. arXiv: 2001.00216 [math.OC].

[DS83] John E. Dennis Jr. and Robert B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations.* Prentice Hall Series in Computational Mathematics. Prentice Hall, Inc., Englewood Cliffs, NJ, 1983, p. 171. ISBN: 0-13-627216-9.

[Fra10a] Rich Franzen. *KODAK Image 09*. http://r0k.us/graphics/kodak/kodim09.html. Accessed: 2021-05-28. 2010.

[Fra10b] Rich Franzen. *KODAK Image 13*. http://r0k.us/graphics/kodak/kodim13.html. Accessed: 2021-05-28. 2010.

[Fra10c] Rich Franzen. *KODAK Image 22*. http://r0k.us/graphics/kodak/kodim22.html. Accessed: 2020-12-11. 2010.

[GK92] Nick Galatsanos and Aggelos Katsaggelos. "Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation". In: *IEEE Transactions on Image Processing* 1.3 (1992), pp. 322–336. DOI: 10.1109/83.148606.

[GTV11] Pascal Getreuer, Melissa Tong, and Luminita A Vese. "A variational model for the restoration of MR images corrupted by blur and Rician noise". In: *International Symposium on Visual Computing*. Springer. 2011, pp. 686–698.

[GN20] Markus Grasmair and Valeriya Naumova. "Multi-parameter approaches in image processing". submitted. 2020.

[HO93] Per Christian Hansen and Dianne O'Leary. "The Use of the L-Curve in the Regularization of Discrete Ill-Posed Problems". In: *SIAM J. Sci. Comput.* 14 (Nov. 1993), pp. 1487–1503. DOI: 10.1137/0914086.

[HS52] Magnus R. Hestenes and Eduard Stiefel. "Methods of conjugate gradients for solving linear systems". In: *J. Research Nat. Bur. Standards* 49 (1952), 409–436 (1953). ISSN: 0160-1741.

[HL01] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of Convex Analysis*. Jan. 2001. ISBN: 978-3-540-42205-1. DOI: 10.1007/978-3-642-56468-0_4.

[Hub+19] Richard Huber et al. "Total generalized variation regularization for multi-modal electron tomography". In: *Nanoscale* 11.12 (2019), pp. 5617–5632. DOI: 10.1039/c8nr09058k.

[JG00] Peter Johnston and Ramesh Gulrajani. "Selecting the corner in the L-curve approach to Tikhonov regularization". In: *IEEE transactions on bio-medical engineering* 47 (Oct. 2000), pp. 1293–6. DOI: 10.1109/10.867966.

[Kno+10] Florian Knoll et al. "Second order total generalized variation (TGV) for MRI". In: *Magnetic Resonance in Medicine* 65.2 (2010), pp. 480–491. DOI: 10.1002/mrm.22595.

[KLV19] Birgit Komander, Dirk A. Lorenz, and Lena Vestweber. "Denoising of image gradients and total generalized variation denoising". In: *J. Math. Imaging Vision* 61.1 (2019), pp. 21–39. ISSN: 0924-9907. DOI: 10.1007/s10851-018-0819-8. URL: https://doi.org/10.1007/s10851-018-0819-8.

[Li+20] Shuaihao Li et al. "Edge-guided second-order total generalized variation for Gaussian noise removal from depth map". In: *Scientific Reports* 10 (2020).

[LP11] Shuai Lu and Sergei V. Pereverzev. "Multi-parameter regularization and its numerical realization". In: *Numer. Math.* 118.1 (2011), pp. 1–31. ISSN: 0029-599X. DOI: 10.1007/s00211-010-0318-3. URL: https://doi.org/10.1007/s00211-010-0318-3.

[Lud20] Martin Ludvigsen. "Parameter selection for total variation regularization with applications in imaging." Master thesis submitted at NTNU. 2020.

[LL19] Alexander Selvikvåg Lundervold and Arvid Lundervold. "An overview of deep learning in medical imaging focusing on MRI". In: *Zeitschrift für Medizinische Physik* 29.2 (2019), pp. 102–127. DOI: 10.1016/j.zemedi.2018.11.002.

[NW06] Jorge Nocedal and Stephen J. Wright. *Numerical optimization.* Second. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2006, pp. 279–283. ISBN: 978-0387-30303-1; 0-387-30303-0.

[Owr17] Brynjulf Owren. "TMA4212 Numerical solution of partial differential equations with finite difference methods". In: (2017). Lecture notes., p. 16.

[ROF92] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. "Nonlinear total variation based noise removal algorithms". In: *Physica D: Nonlinear Phenomena* 60.1 (1992), pp. 259–268. ISSN: 0167-2789. DOI: https://doi.org/10.1016/0167-2789(92)90242-F. URL: http://www.sciencedirect.com/science/article/pii/016727899290242F.

[Sch+09] Otmar Scherzer et al. *Variational methods in imaging.* Vol. 167. Applied Mathematical Sciences. Springer, New York, 2009, pp. xiv+320. ISBN: 978-0-387-30931-6.

[SA10] Theodore Shifrin and Malcolm Adams. *Linear Algebra: A Geometric Approach.* W. H. Freeman, 2010. ISBN: 9781429215213. URL: https://books.google.no/books?id=QwHcZ7cegD4C.

[SC03] David Strong and Tony Chan. "Edge-preserving and scale-dependent properties of total variation regularization". In: *Inverse Problems* 19.6 (Nov. 2003), S165–S187. DOI: 10.1088/0266-5611/19/6/059. URL: https://doi.org/10.1088%2F0266-5611%2F19%2F6%2F059.

[Wan+04] Zhou Wang et al. "Image Quality Assessment: From Error Visibility to Structural Similarity". In: *Image Processing, IEEE Transactions on* 13 (May 2004), pp. 600–612. DOI: 10.1109/TIP.2003.819861.

[WC12] You-Wei Wen and Raymond H. Chan. "Parameter selection for total-variation-based image restoration using discrepancy principle". In: *IEEE Trans. Image Process.* 21.4 (2012), pp. 1770–1781. ISSN: 1057-7149. DOI: 10.1109/TIP.2011.2181401. URL: https://doi.org/10.1109/TIP.2011.2181401.