Denoising and Enhancement of Digital Images

Variational Methods, Integrodifferential Equations, and Wavelets

Dissertation
zur Erlangung des Grades des
Doktors der Naturwissenschaften
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

von
Stephan Didas

Saarbrücken
2008

Tag des Kolloquiums:	14.02.2008
Dekan:	Prof. Dr. Thorsten Herfet
Ausschussvorsitzender:	Prof. Dr. Volker John
Berichterstatter:	Prof. Dr. Joachim Weickert Prof. Dr. Gabriele Steidl Prof. Dr. Xue-Cheng Tai
Protokollführer:	Dr. Bernhard Burgeth

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Saarbrücken, den

Stephan Didas

Kurze Zusammenfassung

Gegenstand der vorliegenden Arbeit sind Verfahren zum Entrauschen, qualitativen Verbessern und Vereinfachen digitaler Bilddaten. Besonderes Augenmerk liegt dabei auf den Beziehungen und der strukturellen Ähnlichkeit zwischen unterschiedlich motivierten Verfahrensklassen. Insbesondere lassen sich die hier behandelten Methoden in drei Klassen einordnen: Bei den Variationsansätzen und partiellen Differentialgleichungen steht der Begriff der Ableitung im Mittelpunkt, um Regularität der Daten und des gewünschten Resultats zu modellieren. Hier wird ein einheitlicher Rahmen für solche Ansätze angegeben, die alle partiellen Ableitungen einer vorgegebenen Ordnung involvieren und experimentell auf stückweise polynomielle Approximationen der gegebenen Daten führen können. Die zweite Klasse von Methoden nutzt Wavelets zur Repräsentation von Daten, mit deren Hilfe sich Filterung als sehr einfache punktweise Anwendung einer nichtlinearen Funktion verstehen lässt. Diese Wavelets als Ableitungen von Glättungskernen aufzufassen bildet die Grundlage für die hier untersuchte Verbindung dieser Verfahren zu Integrodifferentialgleichungen. Im dritten Fall werden Werte des Bildes in einer Nachbarschaft gemittelt, wobei die Gewichtung bei dieser Mittelung adaptiv nach verschiedenen Kriterien angepasst werden kann. Durch Verfeinern des Pixelgitters und Übergang zu Skalierungslimites werden auch hier Verbindungen zu partiellen Differentialgleichungen sichtbar, die in den vorher dargestellten Rahmen eingeordnet werden. Numerische Aspekte beim Vereinfachen von Bildern werden anhand der NDS-Energiefunktion dargestellt, eines einheitlichen Ansatzes, mit dessen Hilfe sich viele der vorgenannten Methoden realisieren lassen. Das Verhalten der einzelnen Filtermethoden wird dabei jeweils durch numerische Beispiele dokumentiert.

Abstract

The topics of this thesis are methods for denoising, enhancement, and simplification of digital image data. Special emphasis lies on the relations and structural similarities between several classes of methods which are motivated from different contexts. In particular, one can distinguish the methods treated in this thesis in three classes: For variational approaches and partial differential equations, the notion of the derivative is the tool of choice to model regularity of the data and the desired result. A general framework for such approaches is proposed that involve all partial derivatives of a prescribed order and experimentally are capable of leading to piecewise polynomial approximations of the given data. The second class of methods uses wavelets to represent the data which makes it possible to understand the filtering as very simple pointwise application of a nonlinear function. To view these wavelets as derivatives of smoothing kernels is the basis for relating these methods to integrodifferential equations which are investigated here. In the third case, values of the image in a neighbourhood are averaged where the weights of this averaging can be adapted respecting different criteria. By refinement of the pixel grid and transfer to scaling limits, connections to partial differential equations become visible here, too. They are described in the framework explained before. Numerical aspects of the simplification of images are presented with respect to the NDS energy function, a unifying approach that allows to model many of the aforementioned methods. The behaviour of the filtering methods is documented with numerical examples.

Zusammenfassung

Gegenstand der vorliegenden Arbeit sind Modelle zum Entrauschen, Vereinfachen und zur qualitativen Verbesserung digitaler Bilddaten. Mit der wachsenden Leistung der vorhandenen Rechner erfreut sich dieses Anwendungsgebiet einer stetig steigenden Relevanz. Dies hat in den letzten Jahrzehnten zu einer starken Forschungsaktivität in diesem Bereich geführt, deren Resultat eine große Anzahl verschiedener Herangehensweisen zur qualitativen Verbesserung von Bildern ist. Besonderes Augenmerk liegt in dieser Arbeit auf den Beziehungen zwischen scheinbar unterschiedlichen Verfahren. Dabei werden im Wesentlichen drei Verfahrensklassen betrachtet: Die erste Klasse bilden Variationsansätze und partielle Differentialgleichungen. Das zentrale mathematische Hilfsmittel sind hierbei Ableitungen. Die zweite Klasse von Verfahren basiert auf der Repräsentation der Daten mit Hilfe von Wavelets. Diese Darstellung erlaubt es, den Filtervorgang als sehr einfache punktweise Anwendung einer nichtlinearen Funktion auf die Wavelet-Koeffizienten durchzuführen. Als dritte und letzte Klasse werden adaptive Mittelungsfilter betrachtet. Hierbei ergibt sich ein Grau- oder Farbwert im gefilterten Bild als gewichtetes Mittel aus allen Werten innerhalb einer Umgebung des betrachteten Pixels. Unterschiedliche Strategien zur Wahl der Gewichte in Abhängigkeit von den vorhandenen Daten in der Umgebung oder im gesamten Bild erlauben diesen sehr anschaulich konstruierten Filtern, qualitativ hochwertige Resultate zu erzielen.

Ziel der Arbeit ist es nun, mit Hilfe der Beziehungen zwischen diesen Methoden das Verständnis für den Filtervorgang zu vertiefen. Über den reinen Erkenntnisgewinn hinaus soll dies dazu genutzt werden, Konzepte von einer Verfahrensklasse auf die andere übertragen zu können und so eine qualitative Verbesserung möglich zu machen. An zentraler Stelle stehen für uns dabei die partiellen Differentialgleichungen: Zum einen wird ein allgemeiner Rahmen für Regularisierung und zugehörige Evolutionsgleichungen angegeben, der auf allen partiellen Ableitungen einer vorgegebenen Ordnung basiert. Interessant ist hierbei, dass die Lösungen experimentell lokal durch Polynome beschrieben werden können, deren Grad sich aus der gewählten Ableitungsordnung ergibt. Zum anderen werden die Wavelet-Verfahren sowie die Mittelungsfilter mit Integrodifferentialgleichugen in Beziehung gebracht. Die hierbei hergeleiteten Gleichungen entsprechen nicht exakt den vorher behandelten Evolutionsgleichungen, was zur Veranschaulichung der Unterschiede im Verhalten der Verfahren beiträgt.

Die vorliegende Arbeit gliedert sich wie folgt in sechs Kapitel: Das erste Kapitel gibt eine Einführung in das Problem sowie den bisherigen Stand der Forschung.

Im zweiten Kapitel werden Methoden betrachtet, die auf Ableitungen der gegebenen Daten oder des gesuchten Resultats beruhen. Dazu gehören Variationsansätze, die unter Verwendung von Ableitungen eine Glattheitsforderung an die Resultate modellieren. Zur Lösung der auftretenden Minimierungsprobleme werden im zweiten Schritt partielle Differentialgleichungen herangezogen. Die be-

schriebenen Verfahren werden im Hinblick auf ihre Stabilitäts- und Skalenraumeigenschaften untersucht.

Das dritte Kapitel beschäftigt sich mit dem Entrauschen durch Wavelet Shrinkage und seiner Formulierung durch Integrodifferentialgleichungen. Die zugrunde liegende Idee dabei ist, die Daten mit Hilfe von Wavelets in eine geeignete Repräsentation zu überführen, in der sich Rauschen durch sehr einfache nichtlineare Transformationen entfernen lässt. Im Falle des klassischen Shrinkage besteht diese Transformation einfach in der punktweisen Anwendung einer Shrinkage-Funktion auf die Wavelet-Koeffizienten. Schlüssel zum Verständnis des Wavelet Shrinkage als Integrodifferentialgleichung ist, die Wavelets als Ableitungen von Glättungskernen aufzufassen. Aus der dadurch entstehenden Vorglättung der Ableitungen ergeben sich die wesentlichen Unterschiede der Wavelet-Verfahren zu den vorher beschriebenen Differentialgleichungen.

Zu den einfachsten und zeitlich am längsten bekannten Methoden zum Entrauschen gehören adaptive Mittelungsverfahren, die in Kapitel vier behandelt werden. Hier geht man von diskreten Formulierungen der Mittelung aus, die direkt die Implementierungen widerspiegeln. Eine Verfeinerung des Pixelgitters und der Übergang zu Skalierungslimites ordnet den Mittelungsverfahren zugehörige partielle Differentialgleichungen zu. In einer Raumdimension ergibt sich dadurch eine Variante der nichtlinearen Diffusion, die eine verbesserte Kantenverstärkung aufweist. Die Verallgemeinerung auf den 2-D-Fall ist nicht eindeutig. Hier werden mehrere Wege beschrieben, die auf anisotrope Diffusionsgleichungen führen. Besonderes Gewicht erhält hier die Untersuchung einer Filtermethode, die eine Verallgemeinerung der Mean Curvature Motion darstellt, jedoch im Gegensatz zur klassischen Gleichung in der praktischen Implementierung visuell scharfe Kanten möglich macht.

Kapitel fünf untersucht numerische Probleme im Zusammenhang mit dem Verbessern von Bildern anhand des NDS-Optimierungsansatzes, der ähnlich wie die adaptiven Mittelungsverfahren eine erweiterte Nachbarschaft eines Pixels beim Entrauschen mit berücksichtigt. Für das bei diesem Ansatz zu lösende Minimierungsproblem wird die Existenz eines Minimums gezeigt. Daneben werden unterschiedliche numerische Methoden zur Approximation eines Minimums erläutert und verglichen.

Eine Zusammenfassung der wesentlichen Resultate sowie ein Ausblick auf interessante Fragestellungen, die sich in diesem Kontext ergeben, bilden mit Kapitel sechs den Abschluss der Arbeit.

Preface

The topics of this thesis are models for denoising, simplification, and qualitative enhancement of digital image data. With the growing calculating and storage capacity of present computers, this area of application enjoys an increasing relevance. This has led to a strong scientific activity whose result is a large choice of different approaches for image enhancement. Special focus in this thesis is on the relations between apparently different methods. In essence, three classes of methods are considered here: The first class is build by variational methods and partial differential equations. The central mathematical tool in this class are derivatives. The second class of methods is based on data representation with the help of wavelets. This representation allows to pursue the filtering as a simple pointwise application of a nonlinear function on the wavelet coefficients. As third and last class, adaptive averaging filters are considered. Here, a grey or colour value in the filtered image is calculated as weighted average out of all values inside a neighbourhood of the pixel under consideration. Different strategies to choose the weights depending on the present data in the neighbourhood or in the whole image allow these very intuitively constructed filters to achieve high quality results.

The goal of the present thesis is to deepen the understanding of the filtering behaviour with the help of the relations of the methods to each other. Beyond the pure gain of insight, this shall be used to carry on concepts from one class of methods to the other and thus to make a qualitative enhancement possible. The partial differential equations are at the central position for us: On the one hand we describe a general framework for regularisation and corresponding evolution equations which is based on all partial derivatives of a given order. An interesting feature of this framework is that experiments show that the solutions can locally be described by polynomials whose degree is determined by the order of the derivatives. On the other hand, the wavelet methods and the adaptive averaging methods are related to integrodifferential equations. The differential equations deduced in this reasoning are not exactly equal to the evolution equations discussed before, which contributes to the illustration of the differences in the behaviour of the methods.

The present thesis consists of six chapters: The first chapter provides an introduction to the problem and the state of the art.

In the second chapter, methods based on derivatives of the given data and the desired result are considered. This encompasses variational methods that model a smoothness demand on the results with the help of derivatives. To solve the occurring minimisation problems, in the second step, partial differential equations are used. The described methods are investigated with respect to their stability and scale-space properties.

The third chapter considers denoising by wavelet shrinkage. Here the basic idea is to transform the data with the help of wavelets into a suitable represen-

tation such that noise can be removed by very simple nonlinear operations. In the case of classical wavelet shrinkage, this operation is just the pointwise application of a nonlinear shrinkage function on the wavelet coefficients. The key to the understanding of wavelet shrinkage as integrodifferential equation is to interpret wavelets as derivatives of smoothing kernels. This models a presmoothing of derivatives from which the fundamental differences between the wavelet-based methods to the previously described differential equations originate.

Adaptive averaging methods belong to the easiest and historically earliest methods known in the context of denoising and are discussed in Chapter 4. Here, one starts with a discrete formulation of the averaging which directly mirrors practical implementations. A refinement of the pixel grid and the transfer to scaling limits maps the averaging methods to corresponding partial differential equations. In 1-D, this yields a variant of nonlinear diffusion filtering with increased edge enhancement. The generalisation to the 2-D case is not unique. Here, several ways will be described which lead to anisotropic diffusion equations. Special focus is on the investigation of a filtering method which can be seen as generalisation of mean curvature motion. In practical implementations, this family can make visually sharp edges possible in opposite to the classical process.

Chapter 5 investigates numerical problems in the context of image enhancement on the basis of the NDS optimisation approach. Similar to the adaptive averaging processes, this approach takes an augmented neighbourhood of the pixels into consideration. We prove the existence of a minimum for the minimisation problem arising here. Further we illustrate and compare several numerical methods for the approximation of this minimum.

A summary of the essential results and an outlook of interesting questions arising in this context build the conclusion of this thesis in Chapter 6.

Acknowledgements

First of all, I would like to thank Prof. Dr. Joachim Weickert for providing me the interesting topic and giving me the opportunity to persue my studies in his group. He has helped me with all the scientific and administrative questions and problems and always encouraged me whenever problems came up. Most of the work presented here was done within the project "Relations Between Nonlinear Filters in Digital Image Processing" which was a part of the priority programme 1114 "Mathematical Methods for Time Series Analysis and Digital Image Processing" financed by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG). My thank goes to the DFG for financing this project and establishing the connections between the participating scientists. I would like to thank Prof. Dr. Gabriele Steidl for giving me the chance to work in this joint project of the groups in Saarbrücken and Mannheim. I am grateful to her for many fruitful discussions. I would like to thank her and Prof. Dr. Xue-Cheng Tai for serving as external reviewers of this thesis.

I am also grateful to Prof. Dr. Jürgen Franke for helpful discussions on the stochastical interpretation of the NDS functional and to Prof. Dr. Roland Duduchava for his interest and his helpful comments on pseudodifferential operators and their regularity properties.

I would like to thank all current and former members of the Mathematical Image Analysis (MIA) group at Saarland University for creating a creative and comfortable atmosphere which made me enjoy working here. In addition my thank goes to all co-authors of joint publications written during my studies. Concerning this thesis, I am grateful to Natalie Marx and Markus Zacharski for proof-reading.

Finally, I would like to thank my parents for their constant support during my studies and my whole life, and my girlfriend Natalie for being with me.

Saarbrücken, February 2008 Stephan Didas

Contents

1 Introduction **5**
- 1.1 Motivation . . . 5
- 1.2 Variational Methods and PDEs . . . 6
  - 1.2.1 Scale-Space Concept and Linear Diffusion . . . 6
  - 1.2.2 Regularisation and Isotropic Nonlinear Diffusion . . . 8
  - 1.2.3 Anisotropic Nonlinear Diffusion . . . 19
  - 1.2.4 Mean Curvature Motion and Related Filters . . . 19
- 1.3 Denoising with Wavelets . . . 22
  - 1.3.1 Wavelet Shrinkage . . . 23
  - 1.3.2 Relations to Nonlinear Diffusion . . . 25
- 1.4 Adaptive Averaging Filters . . . 26
  - 1.4.1 Classical Methods . . . 27
  - 1.4.2 Nonlocal Data and Smoothness Terms . . . 29
  - 1.4.3 Nonlocal Means . . . 29
- 1.5 Outline . . . 30

2 Variational Methods and PDEs **31**
- 2.1 Linear Filtering with Fractional Derivatives . . . 31
  - 2.1.1 Fractional Powers of the Laplacian . . . 32
  - 2.1.2 Regularisation with Fractional Derivative Orders . . . 34
  - 2.1.3 Diffusion with Fractional Derivative Orders . . . 39
  - 2.1.4 Semi-Discrete Linear Filtering . . . 44
  - 2.1.5 Numerical Experiments . . . 46
- 2.2 Nonlinear Filtering with Higher Derivatives . . . 47
  - 2.2.1 Calculus of Variations . . . 48
  - 2.2.2 Higher Order Regularisation and Diffusion . . . 50
  - 2.2.3 Local Feature Enhancement . . . 55
  - 2.2.4 Discrete Nonlinear Filtering . . . 59
  - 2.2.5 Numerical Experiments . . . 67
- 2.3 Higher Order Data Terms . . . 73
  - 2.3.1 Modelling . . . 75
  - 2.3.2 Discretisation and Numerical Experiments . . . 76
- 2.4 Summary . . . 78

3 Integrodifferential Equations **81**
3.1 Relations in the Continuous Setting 81
3.1.1 Wavelet Shrinkage 82
3.1.2 Wavelet Transforms as Smoothed Derivative Operators . . 84
3.1.3 Wavelet Shrinkage and Evolution Equations 86
3.1.4 Variational Methods and Correspondences 88
3.1.5 Pseudodifferential Operators 90
3.2 Relations in the Discrete Setting 94
3.2.1 Description of the Classical Methods 94
3.2.2 Discrete Wavelets and Convolution Kernels 98
3.2.3 Relations Between Both Methods 105
3.2.4 Generalisation to Higher Dimensions 107
3.2.5 Numerical Experiments 110
3.3 Summary 113

4 Adaptive Averaging and PDEs **117**
4.1 Averaging Filters and Scaling Limits in 1-D 117
4.1.1 Derivation of the Scaling Limit 118
4.1.2 Discretisation and Properties 120
4.2 Averaging Filters and Scaling Limits in 2-D 122
4.2.1 Local Averaging and Anisotropic Diffusion 122
4.2.2 Rotational Invariance with the Bilateral Filter 124
4.3 Generalised Mean Curvature Motion 126
4.3.1 Interpretation 126
4.3.2 Choices for the Nonlinearity 127
4.3.3 Discretisation 129
4.4 Numerical Experiments 130
4.5 Summary 133

5 Semilocal Methods, NDS **139**
5.1 The NDS Model 139
5.1.1 The NDS Energy Function 139
5.1.2 Included Classical Methods 141
5.1.3 Parameter Selection 142
5.2 Minimisation Methods 144
5.2.1 Jacobi Method 145
5.2.2 Gauß-Seidel Method 147
5.2.3 Newton's Method 147
5.2.4 Gauß-Seidel Newton Method 148
5.3 Numerical Experiments 148
5.4 Summary 151

6 Conclusions and Outlook **153**
6.1 Conclusions 153
6.2 Outlook 155

A Contributions and Publications **157**
A.1 Further Contributions 157
A.2 Publications 158

Chapter 1

Introduction

Acquisition, storing, and processing of digital images are still becoming more and more important applications in our daily life. Denoising, smoothing, and simplification are central problems in digital image processing. All three of them aim at enhancing the quality of an image either to a human observer or as preprocessing step for a computer vision system. The crucial point in these approaches is to distinguish between important image features that should be kept or even enhanced, and those parts of the image content that are considered as noise and should be removed. Mathematically very different ways have been used to model how a smooth image should look like: For example, a certain smoothness can be formalised in terms of differentiability orders and small modulus of derivatives. This idea leads to regularisation methods and related partial differential equations. Another kind of smoothness assumption is that certain wavelet coefficients should be small, leading to the popular wavelet shrinkage methods. While the preceeding ideas already use formulations with the help of functions, adaptive averaging approaches usually start directly on the level of digital data sampled at discrete pixels. Comparing the differences of the grey value helps to find important features such as edges in this case.

1.1 Motivation

The topics of this thesis are techniques for image simplification and noise removal. For simplification, we investigate techniques to remove details while the important information is kept. We do not specify what is important information here since we do not focus on a specific application. Usually, image edges are considered as important features. We will see that there are also other criteria, for examples large values of higher derivatives or wavelet coefficients. For noise removal, the underlying model for the image degradation is an additive noise model. Thus in the discrete setting, we have three images $f, g, n \in \mathbb{R}^N$ such that

$$f \;=\; g + n \tag{1.1}$$

where f denotes the observed noisy image, g is an unknown ideal image and n is the noise. The task is to recover an image $u \in \mathbb{R}^N$ that is as close as possible to the ideal image g in a certain sense. For some numerical examples to determine the denoising quality, we will use a prescribed g and artificial noise. To measure the distance of the result u to the ideal image g, we usually take the following simple error norms:

$$\|u-g\|_1 := \sum_{j\in J} |u_j - g_j| \quad \text{and} \quad \|u-g\|_2 := \left(\sum_{j\in J} |u_j - g_j|^2\right)^{\frac{1}{2}} \tag{1.2}$$

where $J := \{1, \ldots, N\}$ denotes the set of pixel indices. We do not make any assumptions on the noise distribution except the fact that we usually assume zero mean of the noise. Our focus is on different image denoising and simplification techniques rather than on the physical modelling of noise and of the design of appropriate noise removal strategies for a certain acquisition technique here.

The major goal of this thesis is to investigate the relations between several of the above sketched approaches for image denoising. The knowledge obtained hereby is used to extend the existing filters and to carry over useful filter properties from one class to the other. At central position for the thesis are the derivative-based methods. In this context, an extension of the classical methods to arbitrary order nonlinear regularisation and diffusion will be shown. This will provide the basis to link the wavelet-based methods and the adaptive averaging methods to derivative-based methods. In both cases, we do not obtain exactly the standard diffusion equations, but different variants of it.

The following sections of this chapter will give a short overview over several classes of methods for denoising and simplification that have been proposed in the literature and will be of importance for this thesis. The chapter is concluded with an outline of the thesis in Section 1.5.

1.2 Variational Methods and PDEs

Regularisation and diffusion filtering are frequently used and well-studied methods in image processing. First we start with describing linear filters and the scale-space concept and then turn our attention to nonquadratic regularisation and nonlinear PDE methods.

1.2.1 Scale-Space Concept and Linear Diffusion

The scale-space concept is a popular way to formally and axiomatically describe image simplification. Given an image f, a scale-space is a family of simplified versions $\mathcal{T}_t f$ of the image which depends on one parameter $t \geq 0$. For $t = 0$, we

obtain the original image $\mathcal{T}_0 f = f$, and for the limit $t \to \infty$, it is common to demand that the scale-space approaches the average grey value of f.

Already in the 1960s Iijima [102, 103, 204] has axiomatically described a family of operators to simplify images that satisfies four axioms: linearity, translational invariance, scale invariance, and semigroup property. The most prominent special case of this family is the well-known *Gaussian scale-space*. It has become a popular concept in the western world in the 1980s [211, 112, 116, 80, 171]. It is built by filtering the image with a linear diffusion equation

$$\begin{aligned} \partial_t u &= \Delta u \\ u(\cdot, 0) &= f \end{aligned}$$

or, equivalently, convolve the image with a Gaussian kernel

$$G_\sigma := \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{|x|^2}{2\sigma^2}\right)$$

with standard deviation $\sigma = \sqrt{2t}$. Often this filter family is described on unbounded domains like the real axis $\mathbb{R}$ or on the whole space $\mathbb{R}^n$. Its properties include a maximum-minimum principle, convergence towards the average grey value, translational invariance, and in 2-D, it also satisfies rotational invariance.

One of the most important applications of such scale-spaces is feature detection on multiple scales [117, 118]. Points of interest can be maxima and minima of the image itself or of its derivatives. For example, the gradient magnitude or the zero-crossings of the Laplacian are used to detect edges. It has also been investigated to which extend an image can be reconstructed out of certain features in scale-space [105, 106].

In the meantime, also other linear scale-spaces enjoy a growing popularity: A whole class of linear scale-spaces depending on a fractional order $\alpha > 0$ was axiomatically deduced by Pauwels et al. [153]. Duits et al. [69] further investigated these *α-scale-spaces* where $\alpha \in (0, 1]$ can be interpreted as fractional power of the Laplacian in a pseudodifferential equation creating the scale-space. The restriction on α comes from the demand of a maximum-minimum principle for the resulting filters. The most prominent representative of these linear scale-spaces with fractional order is the *Poisson scale-space* by Felsberg and Sommer [77] and its extension as *monogenic scale-space* [78]. Other recently presented linear scale-spaces include the *relativistic scale-space* [26] and the *Bessel scale-space* [25]. Often the image domain is the complete real axis in scale-space theory, whereas some publications are explicitly treating the bounded domain [67, 76], and others are interested in the discrete setting [116].

We also would like to mention the so-called *inverse scale-space* by Scherzer and Groetsch [164] which is coming from inverse problems. Here one starts with a constant signal for $t = 0$ and adds more and more details until the initial signal appears in the limit for t tending to infinity.

In Chapter 2, we are going to investigate the combination of several fractional powers of the Laplacian for regularisation and generalised diffusion filtering. This can be seen as an extension to the α-scale-space framework.

1.2.2 Regularisation and Isotropic Nonlinear Diffusion

Now we are going to take a look at nonlinear simplification methods. The approaches described here can also be seen in the scale-space context. The most interesting step in contrast to the previous subsection is that the requirement of linearity is dropped. This axiom has rather been introduced to simplify the mathematical treatment. If an image is to be enhanced for a human observer, for example, linearity is not motivated by this application, and thus it makes sense to turn the attention to nonlinear methods.

Regularisation Methods

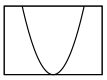
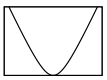
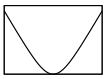
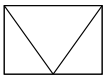
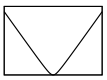
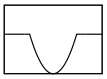
Originally invented as a technique to determine minimum-norm solutions for ill-posed equations by Tikhonov [184], regularisation methods also became popular in image simplification (see [147, 166, 40, 163, 165], for example). A typical energy functional that is minimised for regularisation is given by

$$\mathscr{E}(u) \;=\; \int_\Omega \Big((u-f)^2 \;+\; \alpha\, \Psi(|\nabla u|^2) \Big)\; dx \;. \tag{1.3}$$

Here, $\Omega \in \mathbb{R}^n$ is the image domain and ∇u denotes the image gradient. The first summand is the so-called *data term*. It rewards similarity of the result u with the given data f. For solving an operator equation $\mathcal{A}u = f$ one would replace the data term by $(\mathcal{A}u - f)^2$. This is often applied for the restoration of blurred images under the presence of noise. In the initial regularisation framework (1.3) for denoising, the operator is the identity. The second summand is called *smoothness term* since it penalises deviations from an underlying smoothness assumption. With the smoothness term, one has the possibility to model additional constraints and requirements at the solution. Table 1.1 shows several possible choices for the penaliser functions Ψ. More examples can be found in [55, 145], for example. All penalisers discussed here are continuous and monotonically increasing such that $\Psi(s^2) \leq s^2$ for all $s \in \mathbb{R}$. We notice that not all of the functions shown in Table 1.1 are convex. Strict convexity offers the advantage that the existence and uniqueness of a minimiser can be proven directly. The nonconvex penalisers usually lead to ill-posed problems in a continuous framework [165]. To circumvent this problem, an attempt has been to regularise the derivative in the smoothness term by convolving the image with a Gaussian kernel G_σ of standard deviation σ:

$$\mathscr{E}(u) \;=\; \int_\Omega \big((u-f)^2 + \alpha\, \Psi\left(|\nabla\, G_\sigma * u|^2\right) \big)\; dx \;. \tag{1.4}$$

Table 1.1: Possible choices for penalising functions Ψ.

$\Psi(s^2)$	shape	source
s^2		quadratic regularisation, Tikhonov [184]
$2\lambda^2\left(\sqrt{1+\frac{s^2}{\lambda^2}}-1\right)$		nonquadratic regularisation, Charbonnier et al. [39]
$\log\left(\cosh\left(\frac{s}{\lambda}\right)\right)$		nonquadratic regularisation, Green [91]
$\lvert s\rvert$		total variation, Rudin et al. [158]
$\sqrt{\lambda^2+s^2}-\lambda$		regularised total variation, Acar and Vogel [1]
$\min(s^2,\lambda^2)$		robust statistics, Hampel et al. [97]

This presmoothing makes it possible to show theoretically useful properties like weak lower semicontinuity [165] for nonconvex penalisers such as

$$\Psi(s^2) \;=\; \lambda^2 \log\left(1+\frac{s^2}{\lambda^2}\right) \tag{1.5}$$

related to Perona-Malik diffusion [154]. We will meet this strategy of regularising derivatives with presmoothing in connection with nonlinear diffusion equations again where it has been applied successfully, for example in the filter by Catté et al [32].

Nevertheless, the presmoothed regularisation does not give useful results in practice as examples in [165] show. We will see more examples for this in Chapter 3. Discretising the classical problems (1.3) with finite differences can have such a regularising effect that these methods directly work in practice [202]. The nonconvex penalisers yield edge enhancement and high robustness with respect to noise and outliers which can significantly improve the reconstruction results. Depending on the type of noise or outliers, one can also use a nonquadratic penaliser in the data term. This has shown to be especially helpful for removing impulse noise [146].

Regularisation methods as sketched above are interesting image processing tools since they satisfy some important scale-space properties [165]. For example, they preserve the average grey value of the initial data. Lyapunov functionals and the facts that these methods are translationally and rotationally invariant play also an important role in practice.

Of special interest in the literature is the so-called *total variation regularisation*, or shortly *TV regularisation*, introduced by Rudin, Osher and Fatemi [158]:

$$\mathscr{E}(u) \;=\; \int_\Omega \Big((u-f)^2 \,+\, \alpha\,|\nabla u| \Big)\; dx \;\;. \tag{1.6}$$

This so-called *ROF model* has several interesting properties: As convex energy function it marks the limiting case between the convex and nonconvex regularisers. In this sense, it is a compromise between the aims of achieving robustness against noise and of allowing for a theoretical investigation. From a practical point of view, it is interesting that the regularisation weight is the only parameter of the model. In order to allow for easier numerical implementation, Acar and Vogel [1] have proposed to regularise the penaliser and to use

$$\Psi(s^2) \;=\; \sqrt{\lambda^2 + s^2} - \lambda \tag{1.7}$$

with a small parameter $\lambda > 0$. In the discrete 1-D case, one can see that TV regularisation can be described as a sequence of merging events of neighbouring pixels [176]. There is a finite regularisation weight α for which all pixels have merged, and one obtains the average grey value.

There are many possibilities to solve total variation problems, for example the algorithm by Chambolle [33], or the so-called tube methods [129, 98]. For more information on other applications of TV regularisation, see also [36]. During the last years, also nonlinear inverse scale-space [21] involving total variation has become a field of research.

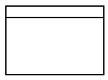
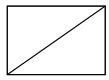
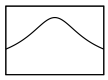
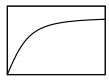
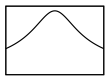
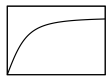
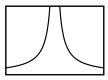
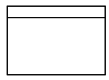
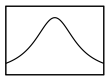
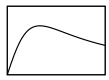
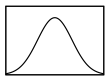
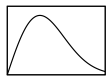
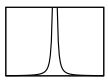
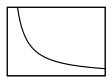
Nonlinear Isotropic Diffusion

In their pioneering work [154], Perona and Malik proposed to use the nonlinear diffusion equation

$$\begin{aligned} \partial_t u &= \operatorname{div}\Big(g(|\nabla u|^2) \nabla u \Big) \\ u(\cdot,0) &= f \end{aligned} \tag{1.8}$$

on the domain $\Omega \subset \mathbb{R}^n$ with homogeneous Neumann boundary conditions $\partial_\nu u = 0$ to simplify images. Here, ν denotes the outer normal of the boundary $\partial\Omega$. The derivative operators ∇ and *div* are concerning only the spatial variables in this context. The initial conditions are the same for all PDE methods in this thesis, and thus they will not be written down explicitly every time. We will see later on that the boundary conditions differ for higher order equations. Possible choices for the diffusivity function g can be found in Table 1.2. Depending on the chosen diffusivity function, one important feature of nonlinear diffusion is that it is able to enhance edges while structures with low contrast are smoothed adaptively. This is explained in terms of forward and backward diffusion [154]: With the help

Table 1.2: Possible choices for the diffusivity function g.

$g(s^2)$	shape	source	flux Φ
1		linear diffusion, Iijima [102], Witkin [211]	
$\left(1+\frac{s^2}{\lambda^2}\right)^{-\frac{1}{2}}$		related to the penaliser by Charbonnier et al. [40]	
$(s^2+\lambda^2)^{-\frac{1}{2}}$		regularised total variation flow, Feng and Prohl [79]	
$\frac{1}{\lvert s\rvert}$		total variation flow, Andreu et al. [4]	
$\left(1+\frac{s^2}{\lambda^2}\right)^{-1}$		nonlinear diffusion, Perona and Malik [154]	
$\exp\left(-\frac{s^2}{2\lambda^2}\right)$		nonlinear diffusion, Perona and Malik [154]	
$\frac{1}{s^2}$		balanced forward-backward diffusion, Keeling and Stollberger [109]	

of the *flux function* $\Phi(s) := g(s^2)s$, the one-dimensional Perona-Malik equation can be written as

$$\partial_t u \;=\; \partial_x\Big(g((\partial_x u)^2)\partial_x u\Big) \;=\; \Phi'(\partial_x u)\partial_x^2 u \;. \tag{1.9}$$

This means that the sign of Φ' determines if we locally have a forward or backward diffusion equation. In Table 1.2, we have displayed the flux functions besides the diffusivities. We see that the upper three flux functions, related to linear, Charbonnier, and regularised total variation diffusion are monotonically increasing. The total variation diffusion has constant flux for nonzero argument. The lower three diffusivities allow for backward diffusion. In particular, the diffusivities

$$g(s^2) \;=\; \frac{1}{1+\frac{s^2}{\lambda^2}} \quad \text{and} \quad g(s^2) \;=\; \exp\left(-\frac{s^2}{2\lambda^2}\right) \tag{1.10}$$

proposed by Perona and Malik [154] are suited for this purpose, since they adaptively allow for forward and backward diffusion. The variable λ plays the role of a contrast parameter to distinguish between regions to be smoothed and edges to be preserved in this case. The balanced-forward-backward diffusivity (BFB) is always in the backward diffusion case. In Chapter 2 we will review this property of nonlinear diffusion filtering in detail and extend it to more general higher order filters.

Even if edge enhancement is a desired property in practice, it can give rise to theoretical difficulties: Kichenassamy [110] has shown that the Perona-Malik filter is not well-posed in the continuous setting. Further interpretations in terms of coupled heat equations or concerning conditions under which comparison principles hold have been given by Esedoglu [73, 74] and Kawohl and Kutev [107].

These ill-posedness problems have led to a regularised version introduced by Catté et al. [32]. It uses a Gaussian kernel to presmooth the argument of the diffusivity:

$$\partial_t u \;=\; \operatorname{div}\left(g\left(|\nabla G_\sigma * u|^2\right)\,\nabla u\right)\;. \tag{1.11}$$

This is sufficient to allow for proving well-posedness properties and also renders the filter more robust against noise. Nevertheless, discrete versions of the original filter (1.8) with finite differences have shown to work well in practice even without this regularisation. A theoretical justification for this fact has been given by Weickert and Benhamouda [202]: The regularising effect of the spatial finite difference discretisation itself is strong enough to make the filter well-posed.

As one can see from the title of the first paper about nonlinear diffusion in image processing by Perona and Malik [154], the nonlinear diffusion filters have been understood in the scale-space context from the beginning. They satisfy some of the most important scale-space properties, for example translational and rotational invariance, preservation of the average grey value, maximum-minimum principle, and Lyapunov functionals [198].

An important special case related to total variation regularisation is

$$g(s^2) \;=\; \frac{1}{|s|}\;, \tag{1.12}$$

which called *total variation flow* or short *TV flow* and has been introduced by Andreu et al. [4]. Steidl et al. [176] have used finite differences to discretise this equation and shown equivalence to total variation regularisation and soft Haar wavelet shrinkage. Feng and Prohl [79] have analysed a finite element discretisation of this problem and introduced

$$g(s^2) = (s^2+\lambda^2)^{-\frac{1}{2}} \tag{1.13}$$

as regularised and bounded version of the diffusivity.

Nonlinear diffusion filtering can be implemented in a very efficient way with the so-called *additive operator splittings (AOS)* by Lu et al. [121] and Weickert et al. [206]. A very simple way to extend the filters to colour images has been proposed by Gerig et al. [87]: All channels are processed by the same equation with a joint diffusivity depending on the sum of the squared gradient norms of all channels. This has been consequently carried over also to the matrix-valued setting [17, 207, 75, 203]. Here, one considers the sum over the squared gradients of all matrix components. In that sense, the matrices entries are treated like multiple channels of a vector-valued image. This does not reflect the special structure

of the matrix-valued data. Nevertheless, there are also promising approaches for generalising nonlinear diffusion equations to matrix-valued data which emphasise the operator-algebraic viewpoint [24, 23, 27]. Here, the derivatives are understood as matrix-valued operations, and also the diffusivity is applied to a matrix in the sense of a functional calculus.

In practical results, there is only one major drawback visible for the Perona-Malik filters: Since the results contain piecewise constant regions, linear grey value transitions in the initial image are hard to recover. Usually the filters tend to oversegment them in constant stairs which gives rise to the name *staircasing* for this artefact. In Figure 1.1 one can see a practical example for staircasing. The stopping time has been chosen such that the ℓ^2-error to the original image was minimised in this case. We have displayed an approximation of the gradient norm, multiplied by a factor of 25 for better visibility. The visualisation of the gradient norm and the sections of the images show clearly some artificial stairs introduced by the second order total variation filter. It is one of the major goals of the nonlinear higher order methods described in Chapter 2 to present a general framework of filtering methods overcoming this drawback.

Relation between Regularisation and Diffusion

Since this will be useful in several contexts later on, we will shortly sketch how regularisation and evolution equations can be related [147, 165].

Our prototype of a variational method for denoising or simplifying an image f can be written as follows: We seek to minimise the energy functional

$$\mathscr{E}(u) \;=\; \int_\Omega (u-f)^2 \, dx \;+\; \alpha \int_\Gamma \Psi(|\mathcal{L}u|^2) \, dy \;. \tag{1.14}$$

We see that it consists of two summands which are weighted by a constant regularisation weight $\alpha \geq 0$. As in (1.3) these two summands can be understood as data and smoothness term. In this more general case, a linear operator $\mathcal{L}$ is used to extract the features of the image u which are relevant for the smoothness assumption. Examples for such operators are derivatives as well as wavelet or Fourier transforms. For the wavelet transform $\mathcal{L} = \mathcal{W}_\psi$, we are going to use this energy functional in Chapter 3. The amount of the extracted features is penalised with a typically nonlinear penalising function Ψ. The variable name y should indicate that the function $\mathcal{L}u$ can have another domain Γ than u has. For example, the wavelet transform $\mathcal{W}_\psi f$ has two variables: The scale and the translation. This will be further specified in Chapter 3. In this way we can use such a variational method for encoding prior knowledge about the image or certain requirements at the solution.

To characterise a minimiser of $\mathscr{E}$, we follow the calculus of variations [86, 5, 178, 88, 7] to obtain a necessary condition on the minimiser u, the so-called

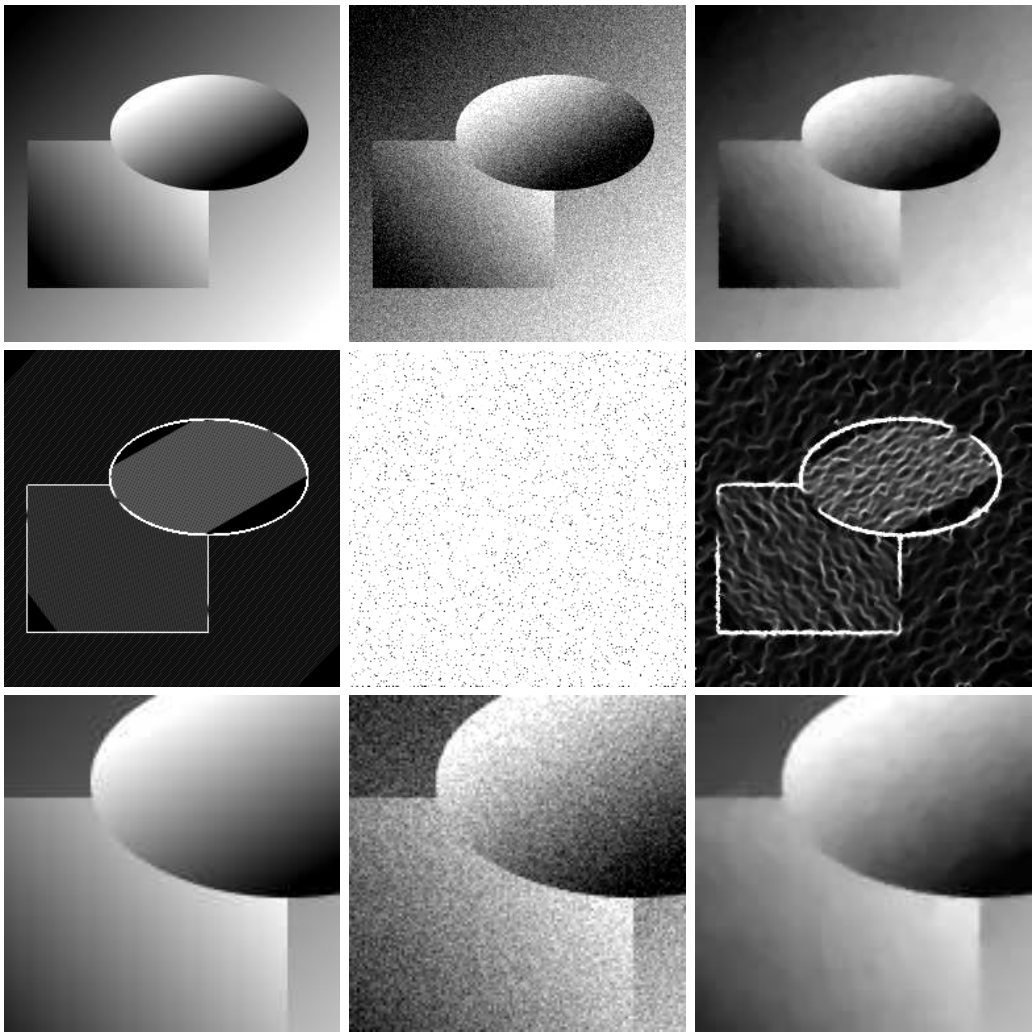


Figure 1.1: *Top left:* Original image, 256×256 pixels. *Top middle:* With additive Gaussian noise, standard deviation $\sigma = 20$. *Top right:* Regularised total variation diffusion $g(s^2) = (s^2 + \lambda^2)^{-1/2}$ with $\lambda = 0.01$, stopping time $t = 20.75$. *Middle row:* Corresponding gradient norm approximations. *Bottom row:* Sections of the images of size 128×128 pixels.

Euler-Lagrange equation

$$0 \;=\; u - f \;+\; \alpha\, \mathcal{L}^* \left(\Psi'(|\mathcal{L}u|^2) \mathcal{L}u \right) \quad . \tag{1.15}$$

Here, $\mathcal{L}^*$ denotes the adjoint operator to $\mathcal{L}$. For the interesting case $\alpha > 0$, this can be written as

$$\frac{u - f}{\alpha} \;=\; -\mathcal{L}^* \left(\Psi'(|\mathcal{L}u|^2) \mathcal{L}u \right) \quad . \tag{1.16}$$

The idea in [165] is now to interpret the left-hand side as approximation of a derivative. To this end we introduce an artificial time variable t in the function u and set $u(\cdot, 0) = f$. Then the right-hand side is a finite difference approximation of the time derivative $\partial_t u(0)$ with step size α. Together, (1.16) can then be considered as an implicit approximation for the evolution equation

$$\partial_t u \;=\; -\mathcal{L}^* \left(\Psi'(|\mathcal{L}u|^2) \mathcal{L}u \right) \quad . \tag{1.17}$$

In the case that $\mathcal{L}$ is the gradient, the adjoint operator is the divergence with a negative sign, and (1.17) coincides with the Perona-Malik equation (1.8). This also explains why the diffusivity function g is often identified with the derivative Ψ' of a penaliser, as we do it for example in the Chapters 2 and 5.

Modifications Leading to Integrodifferential Equations

Besides the pure regularisation and PDE models discussed so far, we want to emphasise the fact that we have already seen some examples for integrodifferential equations. For example, an integration comes into play when a derivative is presmoothed with a convolution kernel. A prominent example for a nonlinear filter using this principle is the regularised nonlinear diffusion (1.11) by Catté et al. [32]. In this example, the presmoothing introduces well-posedness in the problem and makes the filter more robust against noise. Nevertheless, this filter is not directly connected to a variational formulation any more since the inner and outer operators are not adjoint.

An attempt to come close to this filter with variational methods was the presmoothed regularisation (1.4) by Scherzer and Weickert [165]. The corresponding diffusion process to this energy functional is given by

$$\partial_t u \;=\; G_\sigma * \; \operatorname{div} \left(g \left(|\nabla\, G_\sigma * u|^2 \right) \nabla\, G_\sigma * u \right) \quad . \tag{1.18}$$

However, these approaches have hardly found practical applications so far. One problem is that smoothing all derivatives with a fixed scale σ makes the method incapable of removing noise on a smaller scale. This can not happen with the method by Catté et al. since the outer derivatives are not smoothed in this model.

It also has been proposed to replace all derivative operators consequently with Gaussian-smoothed derivatives [144]. The use of presmoothed derivatives

with Gaussian convolution kernels can be seen as one application of the linear scale-space as discussed in Subsection 1.2.1.

In Chapter 3, understanding the wavelet transform as presmoothed derivative operator will play a key role for the connection of wavelet shrinkage to integro-differential equations.

Higher Order Methods

We have already mentioned the staircasing artefacts arising from nonlinear diffusion filters and shown an example in Figure 1.1. The most promising idea to overcome this problem is to introduce higher derivative orders in the filter models [163, 189, 195, 213, 37, 122, 56, 123, 108]. For second or higher order derivatives, one has many different choices how to combine partial derivatives in the smoothness term of the variational method or the nonlinearity of the PDE, respectively.

An extension of classical regularisation by choosing $(\Delta u)^2$ as argument of the penaliser has been proposed by You and Kaveh [213]. With the reasoning given above, such a regularisation is related to evolution equations of the form

$$\partial_t u \;=\; -\Delta\Big(g((\Delta u)^2)\,\Delta u\Big) \;. \tag{1.19}$$

You and Kaveh already mention that equations of this type tend to introduce speckle artefacts around edges. This can be made plausible by the fact that even if $\Delta u = 0$, this does not mean that the second order derivatives have to be zero, and it can come to cancellation effects. Figure 1.2 shows some example for this filtering method where the artefacts are clearly visible. It has been proposed to remove them by postprocessing with a thresholding filter. An approach involving the absolute value of all second order partial derivatives has been proposed by Scherzer [163]. Nevertheless, the resulting method is not rotationally invariant. Taking instead the squared Frobenius norm of the Hessian $\|H(u)\|_F^2$ as penaliser argument solves both problems: It involves all squared partial derivatives of second order and therefore makes cancellation effects impossible, and it is invariant under rotations. This was proposed by Lysaker et al. [122] for the total variation penaliser and is based on an energy functional of the form

$$\mathscr{E}(u) \;=\; \int_\Omega \Big((u-f)^2 + \alpha\|H(u)\|_F\Big)\,dx \;. \tag{1.20}$$

In practice, it yields results without artefacts around the edges, see Figure 1.2, for example. This is the motivation for taking this method as starting point for the framework for higher order regularisation and nonlinear diffusion filters shown in Chapter 2. Since the method of Lysaker et al. [122] is included as a special case of this framework, more details will be given there.

Figure 1.2: Higher order nonlinear diffusion with $g(s^2) = 1/(1 + s^2/\lambda^2)$. *Top left:* Original image, 256×256 pixels. *Top right:* With additive Gaussian noise, standard deviation $\sigma = 10$. *Bottom left:* Method by You and Kaveh [213] without postprocessing, $\lambda = 2.5$, $t = 100$. *Bottom right:* Method by Lysaker et al. [122] $\lambda = 0.01$, $t = 6$.

Methods without Variational Formulation. Two very similar methods without variational formulation have been proposed by Tumblin and Turk [189], and Wei [195]. They use fourth order equations of the form

$$\partial_t u \;=\; -\mathrm{div}\left(g(m)\nabla\Delta u\right) \tag{1.21}$$

where m is a measurement depending on u. In [189], the gradient norm is taken as measurement, while in [195], the Frobenius norm of the Hessian is used. The existence, uniqueness, and properties of solutions for this class of filters are discussed by Greer and Bertozzi in [92]. It should be noted that these methods are not directly related to a variational regularisation approach as we have discussed it before. This comes from the fact that for variational formulations, the corresponding evolution equations always have the adjoint operator outside the nonlinear function as seen above. For the filter (1.21), the divergence as outer derivative is not the adjoint to the (inner) third order derivatives. We will meet filters of this class again in Chapter 3, when we use biothogonal wavelets for shrinkage.

Combinations of Smoothness Terms. Apart from using only classical equations and only higher order methods, it has also been proposed to use a combination of several filtering orders (see [37, 56, 123], for example). Lysaker and Tai [123] used spatially adaptive weights for local convex combinations of second and fourth order terms which gave promising results in practice.

Recently, methods with higher order derivatives have also proved their usefulness for other computer vision applications like optic flow [180, 95, 151] or shape from shading [193].

Pseudodifferential Models

In the last few years, also pseudodifferential operators in the sense of fractional derivative orders have become popular for image filtering. For general information on fractional powers of derivatives and pseudodifferential operators we refer to [148, 64, 181, 90].

Cuesta et al. [50, 51] have introduced a linear equation which yields a transition between the heat and the wave equation. They proposed an equation which can formally be understood as

$$\frac{\partial^\alpha}{\partial t^\alpha} u \;=\; \Delta u \tag{1.22}$$

with $1 \leq \alpha \leq 2$, and so-called semi-linear extensions to it. The diffusion effect is visibly reduced by considering $\alpha \to 2$.

Bai and Feng [9] recently proposed a model for isotropic nonlinear diffusion where not the time derivative, but the spatial derivative orders are real numbers.

They used this model to build the transfer between second and fourth order filtering. Nevertheless, in their model they do not take mixed derivatives into account: They use only higher derivative orders with respect to the coordinate axes' directions.

Pseudodifferential operators have also become popular for other applications such as edge detection [133] or line detection [6]. We are going to use pseudodifferental operators in two contexts in this thesis: Firstly, fractional powers of the Laplacian are building the foundamental tool for the linear scale-spaces considered in Section 2.1. Secondly, power series of derivative operators will allow us to write the wavelet transform without an integral in Section 3.1.

1.2.3 Anisotropic Nonlinear Diffusion

The isotropic filters presented so far have several drawbacks when it comes to the treatment of fiber-like structures or edges. For example, nonlinear isotropic diffusion tends to round edges of objects. To overcome these drawbacks, Weickert has proposed a class of anisotropic filters [198] given by

$$\partial_t u \;=\; \operatorname{div}\Big(D(\nabla\, G_\sigma * u)\,\nabla u\Big)\;. \tag{1.23}$$

Depending on the construction of the diffusion tensor D, it is possible not only to steer the amount of diffusion, but also the direction. This can be used to make the process edge-enhancing or coherence-enhancing, for example (see also [199]).

In this filter family, the presmoothing of the inner derivative is not only used for regularisation, but also necessary for the anisotropy: Due to the construction of the diffusion tensor D, the preferred directions for the anisotropy are the one of $\nabla\, G_\sigma * u$ and its orthogonal direction. Without presmoothing one would simply go in direction of ∇u which directly leads to an isotropic model.

In Chapter 4 we are going to see that some scaling limits coming from adaptive averaging approaches such as bilateral filtering will be anisotropic diffusion filters.

1.2.4 Mean Curvature Motion and Related Filters

Besides anisotropic diffusion, there is a rich class of filters which depend on the local feature directions in the image. Figure 1.3 gives an impression of the properties of several filtering methods.

Following the ideas in [2, 30], one can decompose the Perona-Malik equation (1.8) into two parts acting in orthogonal directions. We consider two relevant local orientations of an image:

$$\eta \;:=\; \frac{\nabla u}{|\nabla u|} \;=\; \frac{1}{\sqrt{u_x^2+u_y^2}}\begin{pmatrix} u_x \\ u_y \end{pmatrix} \tag{1.24}$$

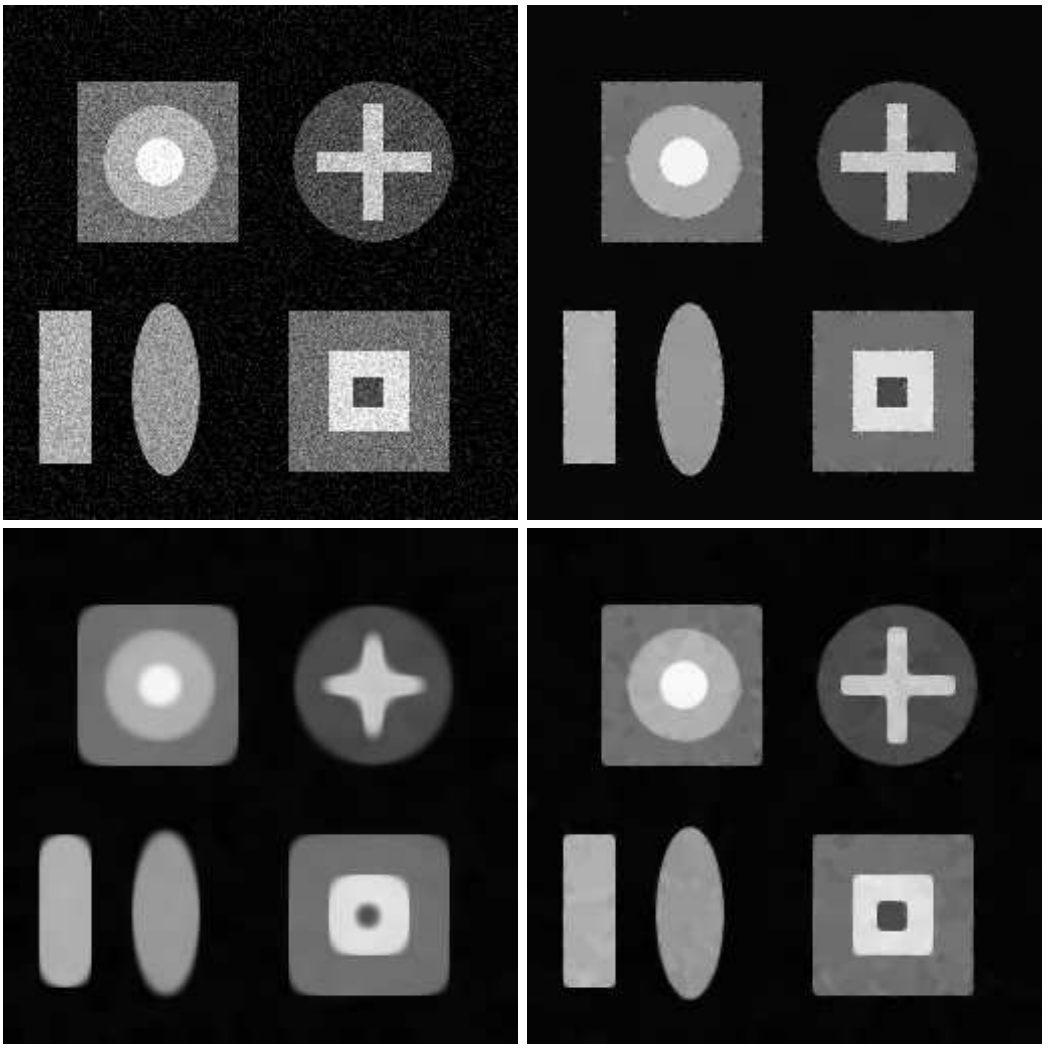


Figure 1.3: Classical PDE-based filters. *Left:* Original image, 256×256 pixels, with Gaussian noise (standard deviation $\sigma = 20$). *Second left:* Perona-Malik filtering, $\lambda = 1$, $t = 50$. *Second right:* Mean curvature motion, $t = 20$. *Right:* Self-snakes, $\lambda = 1$, $t = 50$.

is the direction of the gradient or steepest ascent, that means across an edge in the image. Orthogonal to the gradient we have the direction of the level set

$$\xi \;:=\; \frac{\nabla u^{\perp}}{|\nabla u^{\perp}|} \;=\; \frac{1}{\sqrt{u_x^2+u_y^2}} \begin{pmatrix} -u_y \\ u_x \end{pmatrix} \;, \tag{1.25}$$

which locally points along an edge. In the following considerations we want to decompose a diffusion process into diffusion along and across image edges. For this reason we need the second derivatives of the image in the directions ξ and η, namely

$$u_{\xi\xi} \;=\; \frac{u_{xx}u_y^2 - 2u_x u_y u_{xy} + u_{yy}u_x^2}{u_x^2+u_y^2} \quad \text{and} \tag{1.26}$$

$$u_{\eta\eta} \;=\; \frac{u_{xx}u_x^2 + 2u_x u_y u_{xy} + u_{yy}u_y^2}{u_x^2+u_y^2} \;. \tag{1.27}$$

With these equations we follow Alvarez et al. [2] and decompose the Perona-Malik equation (1.8) into two diffusion components acting in direction ξ and η:

$$\partial_t u \;=\; g(|\nabla u|^2)\, u_{\xi\xi} \;+\; \big(g(|\nabla u|^2) + 2g'(|\nabla u|^2)|\nabla u|^2\big)\; u_{\eta\eta} \;. \tag{1.28}$$

We see that on the one hand, the factor $g(|\nabla u|^2)$ can reduce the velocity of the diffusion close to an edge (when the gradient is large). On the other hand, the first derivative of g in the second summand makes backward diffusion in direction η possible. This gives the filter the capability of edge enhancement.

Starting from (1.28), Carmona and Zhong [30] proposed a more general evolution equation

$$\partial_t u \;=\; c(a u_{\eta\eta} + b u_{\xi\xi}) \tag{1.29}$$

where the function c controls the whole amount of smoothing, and a and b weight this smoothing between the two feature directions. Carmona and Zhong let the functions a, b, and c as given by the Perona-Malik equation (1.28) and focus on different ways to choose the orthogonal set of local feature directions ξ and η. For example, they use eigenvectors of the Hessian $H(u)$ of u or Gabor filters. An alternative framework leading to very similar filters has been proposed by Tschumperlé and Deriche [187, 188]:

$$\partial_t u \;=\; \mathrm{tr}\;(TH(u)) \tag{1.30}$$

with a suitable tensor T. Actually this framework is included in the approach by Charmona and Zhong.

Although this was not mentioned by Carmona and Zhong, mean curvature motion (MCM) can be obtained by choosing special parameters in their general filter class. It only performs smoothing in the direction ξ of the isophotes in the image:

$$\partial_t u \;=\; u_{\xi\xi} \;=\; |\nabla u|\,\mathrm{div}\left(\frac{\nabla u}{|\nabla u|}\right) \;. \tag{1.31}$$

There are some very useful properties of this technique [83, 84, 101] that has been introduced in image processing in [150, 111]: First it is contrast invariant and thus belongs to the morphological operations. Furthermore, it makes nonconvex shapes convex and obeys a shape inclusion principle. Convex shapes are shrunken to circular points and finally vanish. In filtering time $t = \frac{1}{2}r^2$, a circle of radius r and everything inside has vanished. Nevertheless, mean curvature motion has the disadvantage to blur the edges during the evolution.

There are two nonlinear extensions of the MCM method [3, 162] and we want to sketch one of them here: In the context of segmentation, Sapiro [162] proposed the so-called *self-snakes* that can be understood as nonlinear extension of mean curvature motion. The corresponding equation is

$$\partial_t u \;=\; |\nabla u| \mathrm{div}\left(g(|\nabla u|) \frac{\nabla u}{|\nabla u|} \right) \tag{1.32}$$

and allows for sharper edges. This can also be seen at the example in Figure 1.3.

In Chapter 4, we will obtain equations that fit into the framework (1.29) by Carmona and Zhong as scaling limits of adaptive averaging filters. The decomposition into the two feature directions will help us to understand the behaviour of these filters.

1.3 Denoising with Wavelets

The second class of denoising algorithms we are interested in is based on wavelets.

Wavelets allow for a multiresolution representation of data which aims to have a good localisation both in space and frequency. In this sense they can be seen as complement of the spatial representation yielding only spatial information on the one hand and the Fourier representation containing only frequency information on the other hand. An earlier attempt to find a link between both representations was the windowed Fourier transform: It uses Gaussian windows to obtain both localisation in the frequency and the spatial domain and is also called *Gabor transform* [81]. In contrast to the Gabor transform, it is possible to construct bases with scaled and translated versions of the wavelet and avoid redundancy in the representation. Redundancy and missing basis properties are also drawbacks of an approach for multiresolution in image processing, namely the image pyramids [28].

The search for non-redundant basis representation is also the context where the first example for a wavelet appears in the literature: It goes back to the investigation of orthogonal function systems by Haar [96] in 1910. Mainly because of its simplicity, these Haar wavelets are still used very often in image processing even if there have been many different kinds of wavelets proposed in the meantime, for example Daubechies wavelets, symmlets, and coiflets [52, 128].

In electrical engineering, especially in signal and speech processing, the methods of *subband coding* and *filter banks* became popular [48, 49]. Of special interest are perfect reconstruction filter banks with orthogonal filters (introduced by Mintzer [136], Smith, and Barnwell [168]) or biorthogonal filters (Vetterli [191]). The approach has also been extended to multiple dimensions [190]. However, this technique was rather an engineering approach for processing discrete data and not embedded in a sound mathematical theory.

The link between these engineering techniques and the wavelet theory was built by Mallat and Meyer with the concept of multiresolution analysis [126, 127, 134]. With this idea it is possible to perform the wavelet transform in a very efficient way and understand it as perfect reconstruction filter bank. This was one of the milestones on which the popularity of wavelets has its roots, since this made it possible to link several fields and provide fast wavelet transform algorithms with optimal linear complexity. The frame theory gives a mathematical analysis of overcomplete representations (for an overview see [52, 128] and the references therein).

1.3.1 Wavelet Shrinkage

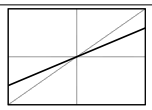
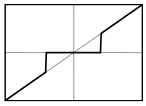
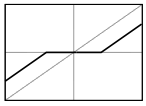
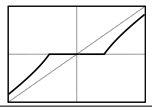
The principle of transforming an image to the wavelet domain and enhancing the quality by shrinking its coefficients has been proposed by Weaver et al. [194] in the context of medical imaging. The notion *wavelet shrinkage* became popular by the papers of Donoho and Johnstone [65, 66]. The general idea is to transform the data to a representation that allows to reduce noise in a straightforward way, namely by diminishing the modulus of the wavelet coefficients. In particular, the low computational complexity of the wavelet transform has made such approaches highly interesting for signal and image processing applications. Obtaining sparse representations by setting small wavelet coefficients to zero can also be used for data compression. This is applied, for example, in the image compression standard JPEG2000 [104].

We will only sketch the idea here, and the mathematical details will be provided in Chapter 3. As already mentioned, wavelet shrinkage consists of three elementary steps:

1. **Analysis:** The data is transformed into a wavelet representation with the help of scaled and translated versions of a mother wavelet. This transform is isometric, and especially there is no information lost.

2. **Shrinkage:** These wavelet coefficients are diminished by pointwise application of a nonlinear shrinkage function $S : \mathbb{R} \longrightarrow \mathbb{R}$. Some examples for typical shrinkage functions can be found in Table 3.1. For a more complete list, see [140], for example.

Table 1.3: Possible choices for the shrinkage function S.

$S(x)$	shape	source
λx		linear shrinkage
$\begin{cases} 0, & \lvert x\rvert \le \lambda \\ x, & \lvert x\rvert > \lambda \end{cases}$		hard shrinkage, Mallat [128]
$\begin{cases} 0, & \lvert x\rvert \le \lambda \\ x - \lambda\mathrm{sgn}(x), & \lvert x\rvert > \lambda \end{cases}$		soft shrinkage, Weaver et al. [194]
$\begin{cases} 0, & \lvert x\rvert \le \lambda \\ x - \frac{\lambda^2}{x}, & \lvert x\rvert > \lambda \end{cases}$		garrote shrinkage, Gao [85]

3. **Synthesis:** The changed coefficients and a synthesis filter are used to construct the output image with the inverse wavelet transform.

As one example for a shrinkage function, we would like to mention the *soft shrinkage* here

$$S(x) := \begin{cases} x - \lambda, & x > \lambda \\ x + \lambda, & x < -\lambda \\ 0, & \text{else} \end{cases} \tag{1.33}$$

since it was introduced first [194] and visualises best the idea of shrinking the coefficients. This means the modulus of all wavelet coefficients is reduced by a given value λ. Other possibilities for shrinkage functions are displayed in Table 1.3.

Wavelet shrinkage in its original form has the serious drawback of lacking translational invariance. A method to overcome this is named *cycle spinning* and has been proposed by Coifman and Donoho [44]. In this approach, translational invariance is introduced by performing wavelet shrinkage on all possible shifted versions of the data. The results are shifted back and averaged afterwards. The computational complexity is reduced by the *algorithme à trous* by Holschneider et al. [99].

To apply wavelet methods in multiple dimensions, usually tensor product wavelets are used [128]. These wavelets have only a limited set of orientations derived from the coordinate directions, and the shrinkage methods do not satisfy rotational invariance. In the discrete 2-D setting, different shrinkage rules for horizontal, vertical and diagonal components can significantly reduce this drawback [137, 138]. Choosing these shrinkage rules was inspired by investigating relations between shrinkage and rotationally invariant nonlinear diffusion filters. Other

approaches for extending the wavelet idea to optimise the representation with respect to orientation or curved edges are, for example, ridgelets and curvelets [29] or orientation scores [68].

Shrinkage as Energy Minimisation

As shown in [34, 35, 14], continuous wavelet shrinkage can be understood as minimisation of certain energy functionals. Of special interest has been soft wavelet shrinkage in this context, which is related to gradient descent along Besov norms.

The discretisation of the scale and the translation parameter allows to work with orthogonal wavelets here. This orthogonality makes it possible to minimise the energy functionals componentwise on all wavelet coefficients.

Hybrid Methods

Besides pure wavelet- and derivative-based approaches, the question how to combine them into hybrid methods has triggered many scientific activities [29, 38, 45, 70, 71, 125]. Since we mainly focus on relations between the two classes, we will not describe details of these methods here.

1.3.2 Relations to Nonlinear Diffusion

During the last years there has been a growing interest in analysing the relations between wavelet-based methods and methods based on partial differential equations (PDEs) such as diffusion filters and their corresponding variational approaches. This includes both works in the continous [10, 34, 35, 42, 43, 53, 135, 172] and in the discrete setting [46, 137, 140, 176, 208, 156]. Let us now focus on some examples that are particularly relevant in the context of this thesis.

As already mentioned above, a variational formulation for wavelet shrinkage has been presented by Chambolle et al. [34, 35] in the continuous setting. This makes it possible to understand wavelet shrinkage as an image smoothing scale-space. Since the scale-space concept originated from diffusion equations [102, 211], this already points out some structural similarities. Nevertheless, a direct comparison between the scale-space properties remained still open. Bredies et al. [14] have shown the equivalence of variational wavelet shrinkage to abstract pseudodifferential evolution equations. One open question in practice is the connection of these pseudodifferential equations to well-known image processing methods such as nonlinear diffusion of Perona-Malik type.

Equivalence results between methods of both classes have also been shown for discrete problems and under certain conditions [176, 140]. For example, discrete soft Haar wavelet shrinkage on the finest scale is equivalent to total variation flow and total variation diffusion [176]. Under more general assumptions, for other

diffusivities or with other kinds of wavelets, one still can show close relationships [140, 205]. These ideas are based on the fact that discrete wavelets on the finest scale approximate derivatives. The derivative order is determined by the number of vanishing moments of the discrete filters. An open question in this context is what happens if we take coarser scales than only the finest one into consideration. In this context only experimental results are available [141]: They indicate that iterations within nonlinear diffusion implementations play a similar role as performing shift-invariant wavelet shrinkage at multiple scales. So far, however, no results have been derived that may be helpful in understanding the differences between both techniques.

1.4 Adaptive Averaging Filters

Adaptive averaging filters belong to the simplest and most effective tools for image processing. Since taking the average of the grey values of all pixels in a certain spatial neighbourhood is an intuitive concept, already early methods in image processing use averaging filters. We can only give an idea of the variety of filters in this context in the following.

In general, a simple averaging filter has the following structure:

$$u_i \;=\; \sum_{j \in J} w(f,i,j)\, f_j \quad \text{for all } i \in J \;. \tag{1.34}$$

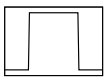
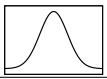
Here, $f \in \mathbb{R}^N$ is the initial discrete image and $u \in \mathbb{R}^N$ denotes its filtered version. The pixels x_i are in a discrete one- or multidimensional domain Ω_h. The index set of these pixels and the corresponding grey values is denoted by $J := \{1, \ldots, N\}$. The weights w depend on the initial data and the two indices i and j. Usually, the weights are assumed to be positive such that $\sum_{j \in J} w(f,i,j) = 1$ for all $i \in J$ in order to have convex combinations of the pixel values. Some of the methods in the literature are used iteratively such that the weights typically depend on the initial data and the result of the last iteration:

$$u^0 \;=\; f \tag{1.35}$$

$$u_i^{k+1} \;=\; \sum_{j \in J} w(u^k, f, i, j)\, u_j^k \;. \tag{1.36}$$

It would also be possible to construct methods which involve also results of previous iteration steps comparable to multistep methods for ordinary differential equations, but here we restrict ourselves to filters involving the initial image and the result from the last step. For example, the discretisation of a nonlinear diffusion filter can be written in such a framework. In general, iterative application of a method with a small neighbourhood can lead to similar results as a single step with a larger neighbourhood, see [141, 155], for example. This will also be the subject of numerical experiments in Chapter 5.

Table 1.4: Possible choices for spatial weights w.

$w(s^2) = \begin{cases} 1, & \lvert s\rvert \le \lambda \\ 0, & \lvert s\rvert > \lambda \end{cases}$		hard window
$w(s^2) = \exp\left(-\frac{s^2}{\lambda^2}\right)$		soft window

Often, adaptive averaging filters are motivated as minimisation of energy functions. This is especially popular in the statistical context where smoothing is motivated as estimation of the true underlying value under the influence of certain noise characteristics. In the following subsections, we will give an overview over methods for choosing the averaging weight w presented in the literature.

1.4.1 Classical Methods

Historically among the first methods in this area are the *sigma filter* by Lee [114] and the *neighbourhood filter* by Yaroslavsky [212]. These methods use two quantities to obtain the weights for averaging: the *tonal* and *spatial distance* between pixels. The tonal distance denotes the difference of the grey values $|f_i - f_j|$ here, while the spatial distance for the corresponding pixels is their distance on the pixel grid $|x_i - x_j|$. The prototype of a sigma filter looks then as follows:

$$u_i = \frac{\sum_{j\in J} w_t(|f_i - f_j|^2) w_s(|x_i - x_j|^2) f_j}{\sum_{j\in J} w_t(|f_i - f_j|^2) w_s(|x_i - x_j|^2)} \tag{1.37}$$

where w_t is the tonal and w_s the spatial weight. Table 1.4 shows the two most important types of spatial weights, defining the so-called *hard* and *soft window*. The only assumption on the spatial weight functions in this thesis will be that they are nonnegative and have a positive value at zero. In general, one could think of negative values in the spatial weight as well: For example, the discretisation of a higher order nonlinear diffusion filter can be understood as adaptive averaging filter involving negative values in its stencil.

A frequently used adaptive averaging method is motivated by statistical estimation: the so-called *spatially weighted M-smoother* [93, 210]. The model is to minimise an energy function of the form

$$E(u) = \sum_{i\in J} \sum_{j\in J} \Psi(|u_i - f_j|^2) w_s(|x_i - x_j|^2) \ . \tag{1.38}$$

This can, for example, be implemented with an iterative scheme

$$u_i^{k+1} = \frac{\sum_{j\in J} w_t(|u_i^k - f_j|^2) w_s(|x_i - x_j|^2) f_j}{\sum_{j\in J} w_t(|u_i^k - f_j|^2) w_s(|x_i - x_j|^2)} \tag{1.39}$$

setting $u^0 = f$. The tonal weight is in this case the derivative of the penaliser $w_t = \Psi'$. The parallels to the sigma filter are obvious.

A very popular method in image processing is the so-called *bilateral filter* of Tomasi and Manduchi [185, 72, 11]. Although it has been presented as a method with one single step, we write it down here in an iterative form:

$$u_i^{k+1} = \frac{\sum_{j\in J} w_t(|u_i^k - u_j^k|^2) w_s(|x_i - x_j|^2) f_j}{\sum_{j\in J} w_t(|u_i^k - u_j^k|^2) w_s(|x_i - x_j|^2)} \quad k \in \mathbb{N} \tag{1.40}$$

starting with $u^0 = f$. It has first been proposed as non-iterative method and thus corresponds to the sigma filter. Altough the filter was becoming popular under the name *bilateral filter*, it has been described before as *nonlinear Gaussian filter* by Aurich and Weule [8] and as *SUSAN* by Smith and Brady [169]. Tomasi and Manduchi proposed weights of the form $\exp(-s^2/\lambda^2)$ for the spatial and tonal weights. This special choice allows for a speedup of the method [152] by understanding the spatial and tonal components as one higher-dimensional vector. This is similar to the idea behind the *Beltrami flow* [170] to consider the image as a surface. Applications of a technique very close to the bilateral filter for other tasks than image denoising are investigated by Smith and Brady with the SUSAN approach [169].

In all these filters, the spatial and tonal weights depend on additional parameters which are chosen globally for the whole image. Sometimes it is hard or even impossible to choose the weights such that semantically important edges are preserved in the whole image while noise is reduced properly. Thus the idea of choosing parameters adaptively depending on local image behaviour came up, which is addressed in the *adaptive weights smoothing* by Polzehl and Spokoiny [157].

Research on relations between neighbourhood filters and PDEs go back to the beginnings of the 1990s [161]. Usually, scaling limits for a spatial and temporal step size tending to zero are used to to build a connection on the way from averaging filters to differential equations [161, 11, 20]. This research direction is somehow natural if one considers similar properties such as edge preservation and enhancement. Furthermore, it is clear that time-explicit discretisations of classical Perona-Malik filters are in fact very special averaging methods. Since they can act edge enhancing, neighbourhood filters can also suffer from staircasing effects. This has been investigated in [20] where a linear regression is proposed to reduce such artefacts.

The first connection of a simple averaging filter named *adaptive smoothing* to Perona-Malik diffusion has been sketched by Saint-Marc et al. [161]. This is also an early example for the attempt to connect averaging methods to nonlinear diffusion filtering which had been proposed only a short time before. For the relations to the Perona-Malik filter, they have considered an averaging filter of the form

$$u_i^{k+1} = c_1 u_{i-1}^k + c_2 u_{i^k} + c_3 u_{i+1}^k \tag{1.41}$$

with $c_1 + c_2 + c_3 = 1$. This normalisation constraint is not used in the derivation of the scaling limit any more. Further, as Barash [11] pointed out, the approximation used in [161] is actually not consistent.

In Chapter 4 we will start with a slightly modified averaging filter to directly involve the normalisation in the scaling limit. Therefore, the derived scaling limit will not be exactly the Perona-Malik equation, but contain an additional factor on the right-hand side coming from the normalisation.

1.4.2 Nonlocal Data and Smoothness Terms

In [139], Mrázek et al. proposed an energy-based approach allowing to consider a whole spectrum of well-known methods as special cases of the same model. This approach makes use of so-called *Nonlocal Data and Smoothness terms*. It will be called *NDS* here. These terms can consider not only information from a small region around a pixel, but can also involve large neighbourhoods. The data term rewards similarity of our filtered image to the given one while the smoothness term penalises high deviations inside a neighbourhood of the evolving image.

To integrate the NDS methods into the framework for neighbourhood filters given above, we will see in Chapter 5 that one way of minimising the function can be written as follows:

$$u^0 = f \tag{1.42}$$

$$u_i^{k+1} = \sum_{j \in J} w_1(u^k, f, i, j)\, u_j^k + \sum_{j \in J} w_2(u^k, f, i, j)\, f_j \ . \tag{1.43}$$

The difference to the iterated average filter (1.35) is that we include some portion of the initial data f in the average in each iteration. This allows for nonflat steady states of the filter for a large class of weight functions w_1 and w_2. One of the major drawbacks of this approach in practice is the relatively high running time. In Chapter 5 we describe several numerical methods for the minimisation of the NDS energy function.

1.4.3 Nonlocal Means

Recently, Buades et al. [18, 19] proposed a new averaging filter which is both appealing through its simple idea and its excellent results: the so-called *nonlocal means*, or shortly *NL-means*. The idea is to determine the weights for an adaptive averaging process by taking similarity of whole neighbourhoods $\mathcal{B}(i) \subseteq J$ and $\mathcal{B}(j) \subseteq J$ of the pixels x_i and x_j into account. Let $f_{|\mathcal{B}(i)}$ denote the entries of f inside the neighbourhood $\mathcal{B}(i)$. The corresponding weights for an NL-means filter are determined as

$$w(f, i, j) = \frac{g(\|f_{|\mathcal{B}(i)} - f_{|\mathcal{B}(j)}\|)}{\sum_{k \in J} g(\|f_{|\mathcal{B}(i)} - f_{|\mathcal{B}(k)}\|)} \ . \tag{1.44}$$

This is especially suitable for the denoising of images which contain repeated image patches, for example in textures. The NL-means approach yields high-quality results. Its major drawback is the high computational complexity since all possible patches or neighbourhoods in the image have to be compared. Ideas for speeding up the process include a pre-selection of patches according to simple pre-computable comparison criteria like the average grey value [124]. If patches differ too much, there is no need to calculate the exact weight, and they are neglected in the averaging step.

NL-means is proposed as one single averaging step, but it is reported to give even higher quality when it is iterated [16]. It has also been shown that only to include the n pixels with the closest patches in the averaging process can increase the quality even further. This number n is typically fixed and small compared to the total number of pixels. In the classical procedure, it can happen that many pixels which have small weights contribute too much to the final result.

The NL-means process will not be directly used in this thesis. Nevertheless, it has been included here as one of the most intersting recent approaches for denoising with averaging filters.

1.5 Outline

So far, we have given an introduction on image processing methods in the literature that are connected to the goals of this thesis.

Let us now take a look at the outline: In Chapter 2 we investigate regularisation and PDE methods involving higher or fractional derivatives of the image. In the first part of the chapter, we describe quadratic regularisation techniques and related linear diffusion methods with linear combinations of fractional derivative orders. We will focus on the scale-space properties of these combinations. The second part will take a closer look at nonlinear methods. We will introduce a general framework for higher-order nonlinear filtering techniques and discuss some of its properties. This framework includes the method by Lysaker et al. [122] as special case. A link between multiscale wavelet methods and integrodifferential equations is build in Chapter 3. We try to characterise the central differences between these integrodifferential equations and classical PDE-based methods as shown in the previous chapter. The considerations will cover both the continuous and the discrete setting. Chapter 4 relates adaptive averaging schemes to nonlinear PDE methods appearing as scaling limits. In two dimensions, there are several possibilities how to generalise the corresponding methods. This will lead to generalised mean curvature motion, a filter family with interesting practical properties. In Chapter 5 we take a closer look at numerical aspects. We investigate several numerical methods for minimising a nonlocal function at the example of he NDS energy function. The thesis is concluded with Chapter 6 where also some interesting questions for further research are pointed out.

Chapter 2

Variational Methods and PDEs

In this chapter we investigate variational methods and related PDE models for denoising and simplification of image data.

In the first section, we focus on quadratic regularisation with linear combinations of fractional powers of the Laplacian and the corresponding pseudodifferential equations. These equations build an extension of the classical scale-space theory as it has been sketched in Section 1.2.1. This section generalises results in [57] to arbitrary spatial dimensions.

Regularisation and PDE methods with integer derivative orders, but nonlinear penalising or diffusivity functions, are considered in the second section. Based on [63], variational methods with higher order smoothness terms are related to a generalisation of the Perona-Malik equation.

The third section discusses some extensions and more general methods. Special emphasis lies on regularisation with derivatives in the data term and the PDE methods originating from this idea.

2.1 Linear Filtering with Fractional Derivatives

We investigate the use of fractional powers of the Laplacian in linear filtering methods for image data. Starting from variational techniques where the smoothness term consists of linear combinations of fractional derivatives, we also describe the corresponding parabolic pseudodifferential equations with constant coefficients. The focus in this description is on the behaviour of the filtering methods in terms of regularity and scale-space properties.

We will see that with the regularisation methods, it is possible to prescribe a certain desired increase in regularity. In contrast to this, the pseudodifferential equations always yield arbitrary smooth solutions and in addition lead to semi-groups. This builds the link to the well-known α-scale-spaces for one single derivative order where we also allow the parameter values $\alpha > 1$ known to violate common maximum-minimum principles. Nevertheless, we show with several

examples that it is possible to construct positivity-preserving combinations of high and low-order filters. Numerical experiments in this direction indicate that non-integer orders play an essential role in this construction. A semi-discrete formulation allows for a very similar formulation to the continuous case and for efficient implementations which make such experiments practicable. In additional numerical experiments we compare the variance decay rates for white noise and edge signals through the action of different filter classes.

This section is organised as follows: In Subsection 2.1.1 we introduce the basic notions related to fractional powers of the Laplacian. Subsection 2.1.2 presents fractional order regularisation as a first application of these notions. The corresponding diffusion equations are described in Subsection 2.1.3. Subsection 2.1.4 reformulates both approaches in a space-discrete framework directly leading to efficient implementations. Our numerical experiments in Subsection 2.1.5 are dedicated in particular to the question of maximum-minimum property and variance decay.

2.1.1 Fractional Powers of the Laplacian

In order to present an elegant concept for fractional powers of the Laplacian, we have to introduce some basic notions first. We consider the n-dimensional case, $n \in \mathbb{N}$. Let us introduce the notations

$$\mathcal{L}^p(\mathbb{R}^n) \;:=\; \left\{ f : \mathbb{R}^n \to \mathbb{R} \;\middle|\; \left(\int_{\mathbb{R}^n} |f(x)|^p \, dx \right)^{\frac{1}{p}} < \infty \right\} \quad \text{for } 1 \le p < \infty$$

and

$$\mathcal{L}^\infty(\mathbb{R}^n) \;:=\; \{ f : \mathbb{R}^n \to \mathbb{R} \mid \operatorname{ess\,sup} f < \infty \} \quad .$$

We denote the Lebesgue spaces where the kernel of the corresponding seminorm is factorised out with $\mathscr{L}^p(\mathbb{R}^n)$. Let $\|f\|_p$ be the norm for $f \in \mathscr{L}^p(\mathbb{R}^n)$. The Fourier transform of a function $f \in \mathscr{L}^1(\mathbb{R}^n)$ is pointwise defined by

$$\hat{f}(\xi) \;:=\; (2\pi)^{-\frac{n}{2}} \int_{\mathbb{R}^n} f(x) \exp(-i\langle x, \xi \rangle) \, dx$$

for $\xi \in \mathbb{R}^n$. Let $\mathcal{F} : \mathscr{L}^2(\mathbb{R}^n) \longrightarrow \mathscr{L}^2(\mathbb{R}^n)$ denote the Fourier-Plancherel transform, i. e. the extension of the mapping $\mathscr{L}^1(\mathbb{R}^n) \ni f \longmapsto \hat{f}$ onto $\mathscr{L}^2(\mathbb{R}^n)$. It is well-known that $\mathcal{F}$ is isometric with respect to the norm in $\mathscr{L}^2(\mathbb{R}^n)$ (see [159] for details). For a multi-index $\gamma \in \mathbb{N}^n$, we define

$$\xi^\gamma := \xi_1^{\gamma_1} \cdot \ldots \cdot \xi_n^{\gamma_n} \,, \quad |\gamma| := \sum_{k=1}^{n} \gamma_k \,, \text{ and } \mathcal{D}^\gamma f := \partial_{x_1}^{\gamma_1} \ldots \partial_{x_n}^{\gamma_n} f \tag{2.1}$$

under the assumption that the corresponding derivatives exist. Later on we will especially make use of the property

$$i^{|\gamma|} \xi^\gamma \mathcal{F} f \;=\; \mathcal{F}\left(\mathcal{D}^\gamma f\right) \tag{2.2}$$

which builds the link between differentiation in the spatial domain and multiplication in the Fourier domain. For $f \in \mathscr{L}^\infty(\mathbb{R}^n)$ let $\mathcal{M}_f : \mathscr{L}^2(\mathbb{R}^n) \longrightarrow \mathscr{L}^2(\mathbb{R}^n)$ denote the multiplication operator defined by $\mathcal{M}_f g := fg$. With this notation (2.2) reads as $\mathcal{M}_{i^{|\gamma|}\xi^\gamma}\mathcal{F}f = \mathcal{F}(\mathcal{D}^\gamma f)$. As we are going to write the linear filtering operations in this section as multiplications with bounded functions in the Fourier domain, let us first consider some useful properties of these multiplication operators:

Lemma 2.1 (Properties of Multiplication Operators)
For $f, g \in \mathscr{L}^\infty(\mathbb{R}^n)$ the multiplication operator $\mathcal{M}_f$ is continuous with respect to the $\mathscr{L}^2(\mathbb{R}^n)$-norm with $\|\mathcal{M}_f\| \leq \|f\|_\infty$. Further, $fg \in \mathscr{L}^\infty(\mathbb{R}^n)$ and $\mathcal{M}_f\mathcal{M}_g = \mathcal{M}_{fg}$.

Proof: With $f \in \mathscr{L}^\infty(\mathbb{R}^n)$ we can write

$$\|M_f h\|_2^2 \;=\; \int_{\mathbb{R}^n} |fh|^2 dx \;\leq\; \sup_{x\in\mathbb{R}^n} |f(x)|^2 \|h\|_2^2 \;=\; \|f\|_\infty^2 \|h\|_2^2$$

for all $h \in \mathscr{L}^2(\mathbb{R}^n)$, and thus we have $\|\mathcal{M}_f\| \leq \|f\|_\infty$. The other two properties follow directly from the definitions. □

Following the notation in [182] we define the Sobolev space

$$\mathscr{H}^s(\mathbb{R}^n) \;:=\; \left\{ u \in \mathscr{L}^2(\mathbb{R}^n) \;\middle|\; \left(1 + |\xi|^2\right)^{\frac{s}{2}} \hat{u} \in \mathscr{L}^2(\mathbb{R}^n) \right\} \tag{2.3}$$

of all functions in $\mathscr{L}^2(\mathbb{R}^n)$ and $s \in \mathbb{R}$. For a vector $\xi \in \mathbb{R}^n$, $|\xi|^2 := \sum_{k=1}^n \xi_k^2$ denotes the squared Euclidean norm here. For natural numbers $s \in \mathbb{N}$, functions in $\mathscr{H}^s(\mathbb{R}^n)$ are weakly differentiable up to the order s. From (2.2) we deduce the spectral decomposition of the negated Laplacian $-\Delta = \mathcal{F}^{-1}\mathcal{M}_{|\xi|^2}\mathcal{F}$. Since the Laplace operator is negative semidefinite, we consider the positive semidefinite operator $-\Delta$ here. This allows us to define fractional powers

$$(-\Delta)^\alpha \;=\; \mathcal{F}^{-1}\mathcal{M}_{|\xi|^{2\alpha}}\mathcal{F} \qquad (\alpha > 0) \tag{2.4}$$

as multiplication operators in the Fourier domain (see [183, 181] for further details).

Lemma 2.2 (Order of Differentiability)
Applying $(-\Delta)^\alpha$ to functions in a certain Sobolev space $\mathscr{H}^s(\mathbb{R}^n)$ with $s > 2\alpha$ reduces the order of differentiability by 2α, i. e.

$$(-\Delta)^\alpha : \mathscr{H}^s(\mathbb{R}^n) \longrightarrow \mathscr{H}^{s-2\alpha}(\mathbb{R}^n) \;. \tag{2.5}$$

Proof: Choose $f \in \mathscr{H}^s(\mathbb{R}^n)$, then we know

$$(1 + |\xi|^2)^{\frac{s}{2}} \hat{f} \in \mathscr{L}^2(\mathbb{R}^n) \quad\Longleftrightarrow\quad |\xi|^s \hat{f} \in \mathscr{L}^2(\mathbb{R}^n) \;. \tag{2.6}$$

With the definition of fractional powers of the Laplacian given above, we obtain

$$\begin{aligned}(-\Delta)^{\alpha} f \in \mathscr{H}^{s-2\alpha}(\mathbb{R}^n) &\iff |\xi|^{s-2\alpha}\left(|\xi|^{2\alpha}\hat{f}\right) \in \mathscr{L}^2(\mathbb{R}^n) \\ &\iff |\xi|^{s}\hat{f} \in \mathscr{L}^2(\mathbb{R}^n)\ .\end{aligned}$$

This shows how the order of differentiability decreases by applying the fractional derivative order. □

In the next sections we are going to replace derivative operators in classical image processing approaches with operators of the type $(-\Delta)^{\alpha}$ and investigate the properties of the resulting filter methods.

2.1.2 Regularisation with Fractional Derivative Orders

To extend linear regularisation to fractional derivative orders we consider the energy functional

$$\mathscr{E}(u) \;=\; \int_{\mathbb{R}^n} \left((u-f)^2 + \sum_{k=1}^{m} \beta_k \left((-\Delta)^{\alpha_k} u\right)^2 \right) dx\ . \tag{2.7}$$

It consists of a quadratic data term, and the smoothness term is a linear combination of $m \in \mathbb{N}$ fractional derivatives. The fractional derivative orders are given as $\alpha = (\alpha_1, \ldots, \alpha_m)^T \in \mathbb{R}_+^m$, and the regularisation weights are denoted with $\beta = (\beta_1, \ldots, \beta_m)^T \in \mathbb{R}_+^m$. Since the regulariser terms are squared, we can simply work with the negated Laplacian here and do not need an additional minus sign. For integer derivative orders α_k, similar functionals have been considered by Nielsen et al. [143]. We assume that the signals u and f may only attain real values. With the Plancherel identity we can rewrite functional (2.7) in the Fourier domain as

$$\mathscr{E}(\hat{u}) \;=\; \int_{\mathbb{R}^n} \left(\left|\hat{u} - \hat{f}\right|^2 + \sum_{k=1}^{m} \beta_k |\xi|^{2\alpha_k} |\hat{u}|^2 \right) d\xi \tag{2.8}$$

depending on the complex Fourier transform $\hat{u}$. A decomposition into the real and imaginary part shows that it is necessary for a minimiser u to satisfy the Euler-Lagrange equation

$$\hat{u} - \hat{f} + \sum_{k=1}^{m} \beta_k |\xi|^{2\alpha_k} \hat{u} \;=\; 0 \quad \text{for all } \xi \in \mathbb{R}^n\ . \tag{2.9}$$

We deduce that the minimiser u of the functional $\mathscr{E}$ has the Fourier transform

$$\hat{u} \;=\; \left(1 + \sum_{k=1}^{m} \beta_k |\xi|^{2\alpha_k}\right)^{-1} \hat{f} \quad \text{for all } \xi \in \mathbb{R}^n\ . \tag{2.10}$$

To obtain a regularised version of f we transform this minimiser $\hat{u}$ in the spatial domain which motivates the following definition:

Definition 2.3 (Fractional Order Regularisation)
For $\alpha = (\alpha_1, \ldots, \alpha_m)$, $\beta = (\beta_1, \ldots, \beta_m) \in \mathbb{R}^m_+$ we denote the multipliers appearing in (2.10) by

$$r^\alpha_\beta : \mathbb{R}^n \longrightarrow \mathbb{R}, \qquad r^\alpha_\beta(\xi) := \left(1 + \sum_{k=1}^m \beta_k |\xi|^{2\alpha_k}\right)^{-1} \tag{2.11}$$

and use these functions to define the regularisation operators

$$\mathcal{R}^\alpha_\beta : \mathscr{L}^2(\mathbb{R}^n) \longrightarrow \mathscr{L}^2(\mathbb{R}^n), \qquad \mathcal{R}^\alpha_\beta = \mathcal{F}^{-1} \mathcal{M}_{r^\alpha_\beta} \mathcal{F} \ . \tag{2.12}$$

First we assure ourselves that the above definition leads to a continuous operator. Furthermore we give a measure for the increase of smoothness obtained by applying a regularisation operator of this class.

Proposition 2.4 (Stability and Regularity of Regularisation)

1. *The regularisation operator $\mathcal{R}^\alpha_\beta$ is continuous with respect to the norm in $\mathscr{L}^2(\mathbb{R}^n)$ with $\|\mathcal{R}^\alpha_\beta\|_2 \leq 1$.*

2. *Regularisation $\mathcal{R}^\alpha_\beta$ increases the smoothness order by twice the maximal derivative order used in the smoothness term:*
   For all $s > 0$, $\mathcal{R}^\alpha_\beta : \mathscr{H}^s(\mathbb{R}^n) \longrightarrow \mathscr{H}^{s+2\alpha^*}(\mathbb{R}^n)$ with $\alpha^* := \max\limits_{k=1,\ldots,m} \alpha_k$.

Proof:

1. The Fourier multipliers satisfy $0 \leq r^\alpha_\beta(\xi) \leq 1$ for all $\alpha, \beta \in \mathbb{R}^m_+$ and all $\xi \in \mathbb{R}^n$, i. e. $\|r^\alpha_\beta\|_\infty \leq 1$. Lemma 2.1 then shows that $\left\|\mathcal{M}_{r^\alpha_\beta}\right\| \leq 1$ and

   $$\|\mathcal{R}^\alpha_\beta\| \ \leq \ \left\|\mathcal{F}^{-1}\right\| \left\|\mathcal{M}_{r^\alpha_\beta}\right\| \|\mathcal{F}\| \ \leq \ 1 \tag{2.13}$$

   using the fact that the Fourier transform is $\mathscr{L}^2$-isometric.

2. Fix $f \in \mathscr{H}^s(\mathbb{R}^n)$. To characterise the Sobolev spaces, we again use (2.6). Let k^* be the index of the maximal exponent α^*. Then we can see that

   $$\left(1 + \sum_{k=1}^m \beta_k |\xi|^{2\alpha_k}\right)^{-1} |\xi|^{2\alpha^*} \ = \ \left(\frac{1}{|\xi|^{2\alpha^*}} + \sum_{\substack{k=1 \\ k \neq k^*}}^m \beta_k \frac{|\xi|^{2\alpha_k}}{|\xi|^{2\alpha^*}} + \beta_{k^*}\right)^{-1} \tag{2.14}$$

   is a bounded factor, and thus it follows that

   $$\left(1 + \sum_{k=1}^m \beta_k |\xi|^{2\alpha_k}\right)^{-1} |\xi|^{s+2\alpha^*} \hat{f} \ \in \ \mathscr{L}^2(\mathbb{R}^n) \tag{2.15}$$

   which implies $\mathcal{R}^\alpha_\beta f \in \mathscr{H}^{s+2\alpha^*}(\mathbb{R}^n)$. □

For integer derivative orders a corresponding statement to the second part of the previous lemma has been shown by Nielsen et al. [143]. As they state for integer orders, also fractional order regularisation is not a projection operator: Applying regularisation iteratively increases the smoothness in each step by twice the maximal derivative order α^*. Starting with a function in $\mathscr{L}^2(\mathbb{R}) = \mathscr{H}^0(\mathbb{R}^n)$ we now are able to reach a given degree of smoothness with linear regularisation. This smoothness property does not depend on the size of the regularisation weights $\beta_k > 0$. The multiplication in the Fourier domain is equivalent to a convolution in the spatial domain.

Several examples for the Fourier multipliers r_β^α and the corresponding convolution kernels are displayed in the Figures 2.1, 2.2, and 2.3. In Figure 2.1 we show the behaviour for one single filtering order, fixed weight $\beta = 1$, and varying orders α. One can see that the kernels for $\alpha = 0.5$ (corresponding to the Poisson scale-space) and for $\alpha = 1$ (Gaussian scale-space) are positive, while a larger α introduces negative parts. This reflects that the maximum-minimum principle is violated for $\alpha > 1$. The fact that the convolution kernel for $\alpha = 2.0$ reaches negative values indicates that the corresponding filter violates a maximum-minimum property. Figure 2.2 again shows the case $\alpha = 2$, but now with changing weight β. The stronger smoothing properties can be seen by the wider kernels in this case. A combination of two filter orders is displayed in Figure 2.3. This is an example that combining a filter of an order $\alpha_1 < 1$ with one of order $\alpha_2 > 1$ can yield positive convolution kernels and thus lead to a scale-space with maximum-minimum principle. This will be the topic of further numerical experiments in Subsection 2.1.5.

Besides its smoothing behaviour the linear filtering technique is also expected to satisfy some scale-space properties. We summarise these in the case of fractional order regularisation:

Proposition 2.5 (Properties of Regularisation)
The regularisation operators $\mathcal{R}_\beta^\alpha$ are linear, invariant under translations and rotations, and preserve the average grey value, i. e.

$$\int_{\mathbb{R}^n} \left(\mathcal{R}_\beta^\alpha f\right)(x) dx \;=\; \int_{\mathbb{R}^n} f(x) dx \;. \tag{2.16}$$

Proof: For the translational invariance we note that translations correspond to multiplications with phase factors $\exp(i\langle c, \xi\rangle)$ with a suitable constant $c \in \mathbb{R}^n$ of absolute value one in the Fourier domain. Since the multipliers r_β^α only assume real values these do not affect the argument of the Fourier coefficients and thus do not interfere with the complex phase factors.

We notice that the rotation of a function and the Fourier transform commute. The functions r_β^α do only depend on $|\xi|^2$, and thus they are rotationally symmetric or radial functions. Together this means that the regularisation operator $\mathcal{R}_\beta^\alpha$ and a rotation commute: The operator is rotationally invariant.

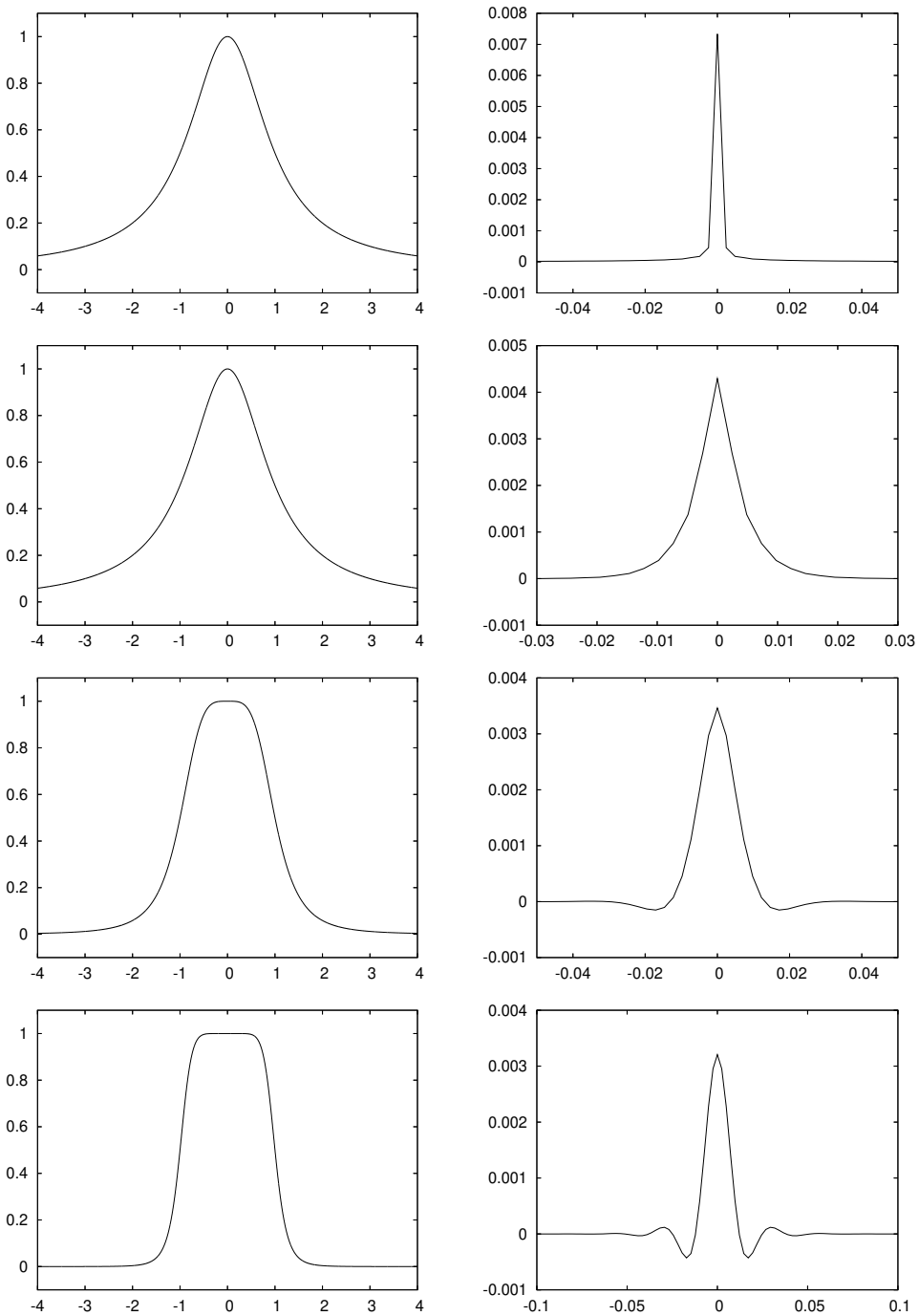


Figure 2.1: Fourier multipliers and corresponding convolution kernels for fractional order regularisation in the one-dimensional case and for one single order α with $\beta = 1$. *Left:* Fourier multipliers r_β^α. *Right:* Corresponding convolution kernels. *From top to bottom:* $\alpha = 0.5, 1, 2, 4$.

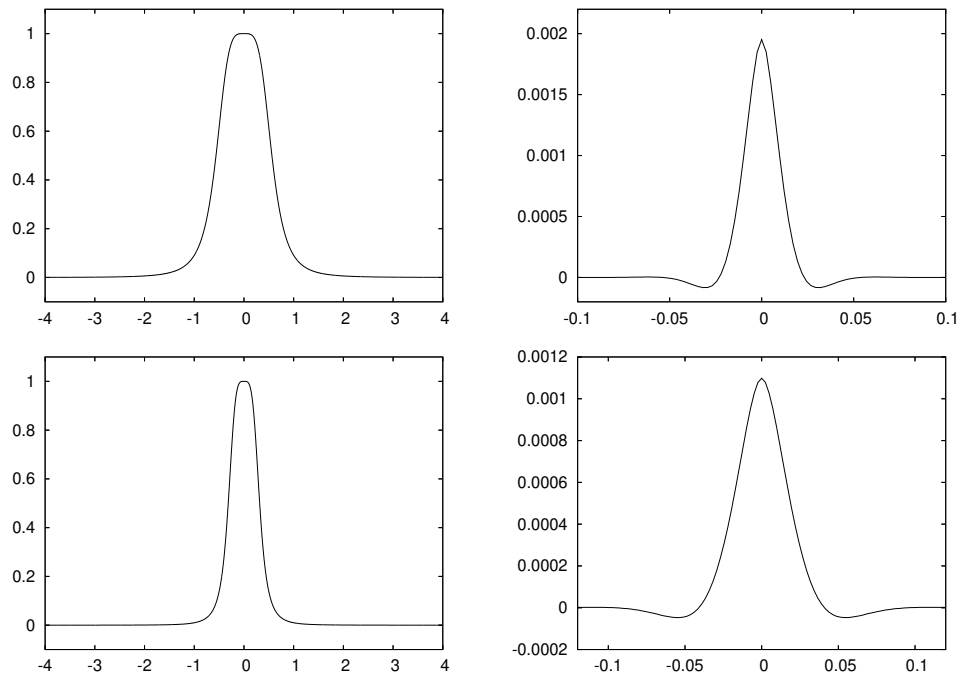


Figure 2.2: Behaviour of the multipliers and convolution kernels for fractional order regularisation with order $\alpha = 2$ for different β. *Left:* Fourier multipliers r_β^α. *Right:* Corresponding convolution kernels. *Top:* $\beta = 10$. *Bottom:* $\beta = 100$.

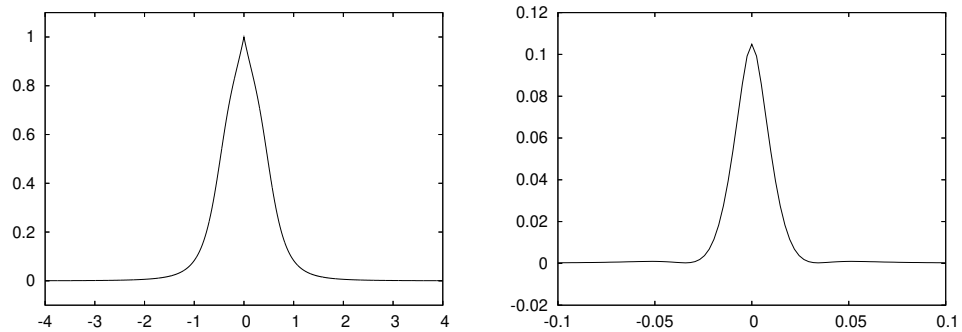


Figure 2.3: Positive combination of two kernel orders $\alpha_1 = 0.5$ and $\alpha_2 = 2$ with weights $\beta_1 = 1$ and $\beta_2 = 10$. *Left:* Fourier multiplier r_β^α. *Right:* Corresponding convolution kernel.

The average grey value can be expressed as $\hat{f}(0) = \int_{\mathbb{R}^n} f(x) \exp(-i\langle x, 0\rangle) dx$. Since $r_\beta^\alpha(0) = 1$ for all $\alpha, \beta \in \mathbb{R}_+^m$, the average grey value remains unchanged by multiplication with r_β^α in the Fourier domain. □

2.1.3 Diffusion with Fractional Derivative Orders

The elliptic differential equations appearing in regularisation techniques are related to parabolic diffusion equations as already described in Subsection 1.2.2. Now we investigate such parabolic equations involving a linear combination of different fractional powers of the Laplacian. To this purpose we choose fractional derivative orders $\alpha_1, \ldots, \alpha_m > 0$ and weight parameters $\lambda_1, \ldots, \lambda_m > 0$ and consider the linear pseudodifferential equation

$$\partial_t u \;=\; -\sum_{k=1}^{m} \lambda_k (-\Delta)^{\alpha_k} u \tag{2.17}$$

with initial condition $u(x, 0) = f(x)$ for all $x \in \mathbb{R}^n$. In the Fourier domain (2.17) reads as

$$\partial_t \hat{u} \;=\; -\sum_{k=1}^{m} \lambda_k |\xi|^{2\alpha_k} \hat{u} \;\; .$$

This is an ordinary differential equation with parameter ξ and can be analytically solved by

$$\hat{u}(\xi, t) \;=\; \exp\left(-t \sum_{k=1}^{m} \lambda_k |\xi|^{2\alpha_k}\right) \hat{f} \;=\; \prod_{k=1}^{m} \exp\left(-t\lambda_k |\xi|^{2\alpha_k}\right) \hat{f} \;\; . \tag{2.18}$$

This formula expresses fractional order linear diffusion filtering as multiplication in the Fourier domain. In the following, we define the corresponding convolution kernels in the spatial domain:

Definition 2.6 (Multipliers and Convolution Kernels for Diffusion)
For the order $\alpha > 0$, the weight $\lambda > 0$ and the stopping time $t \geq 0$, we define the multiplier function

$$G_\lambda^\alpha(\xi, t) \;:=\; \exp\left(-t\lambda |\xi|^{2\alpha}\right) \quad \textit{for all } \xi \in \mathbb{R}^n \;\; .$$

We also define the convolution kernels appearing in linear filtering as the Fourier backtransform

$$p_\lambda^\alpha(x, t) \;:=\; \frac{1}{\sqrt{2\pi}} \mathcal{F}^{-1}\left(G_\lambda^\alpha(\cdot, t)\right)(x) \;=\; \frac{1}{2\pi} \int_{\mathbb{R}} \exp\left(-t\lambda |\xi|^{2\alpha} + i\langle x, \xi\rangle\right) d\xi \;\; .$$

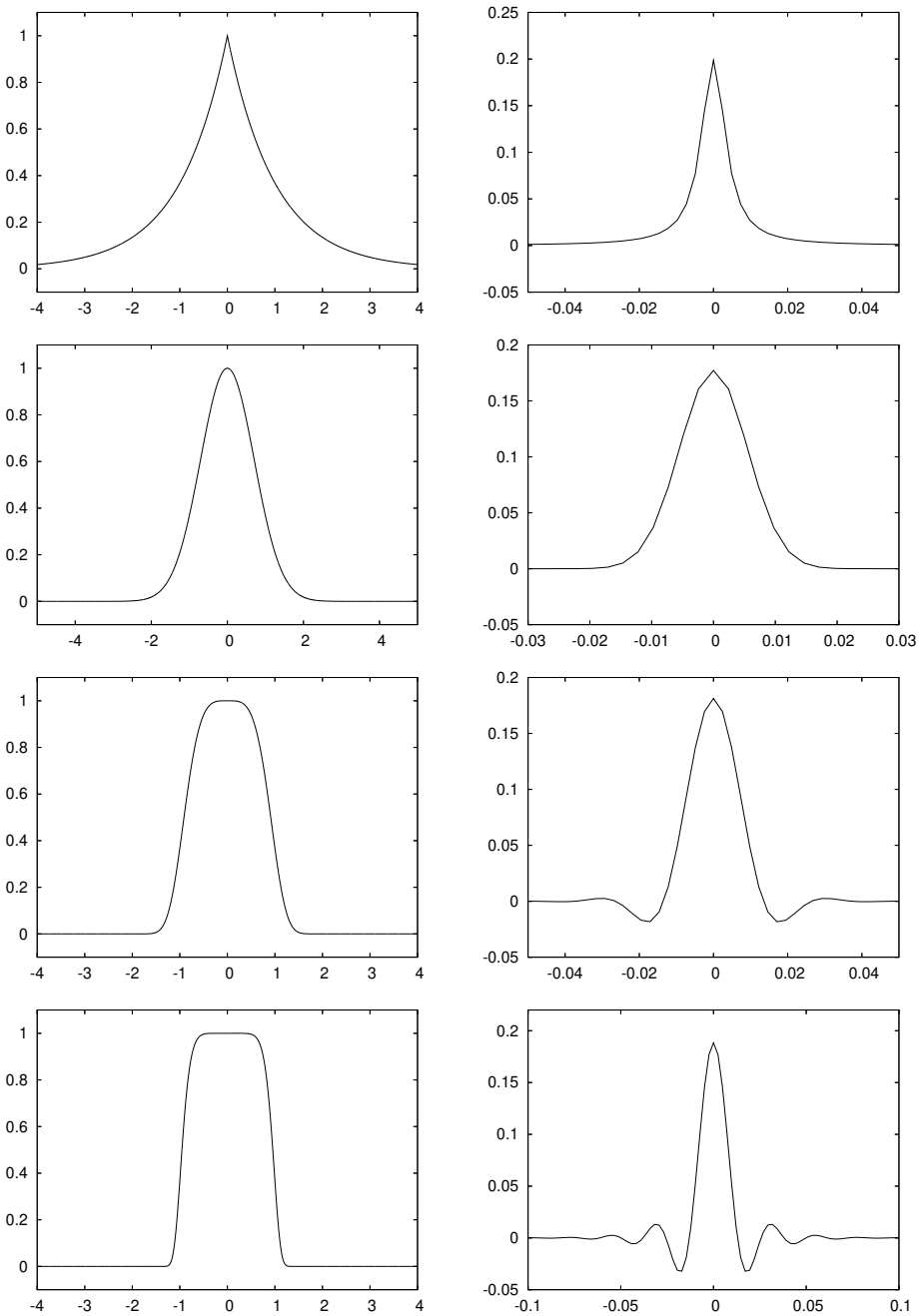


Figure 2.4: Fourier multipliers and corresponding convolution kernels for fractional order diffusion filtering in the one-dimensional case $n = 1$ and with one single filtering order α for $\lambda = 1$ and $t = 1$. *Left:* Fourier multipliers G_λ^α. *Right:* Corresponding convolution kernels p_λ^α. *From top to bottom:* $\alpha = 0.5, 1, 2, 4$.

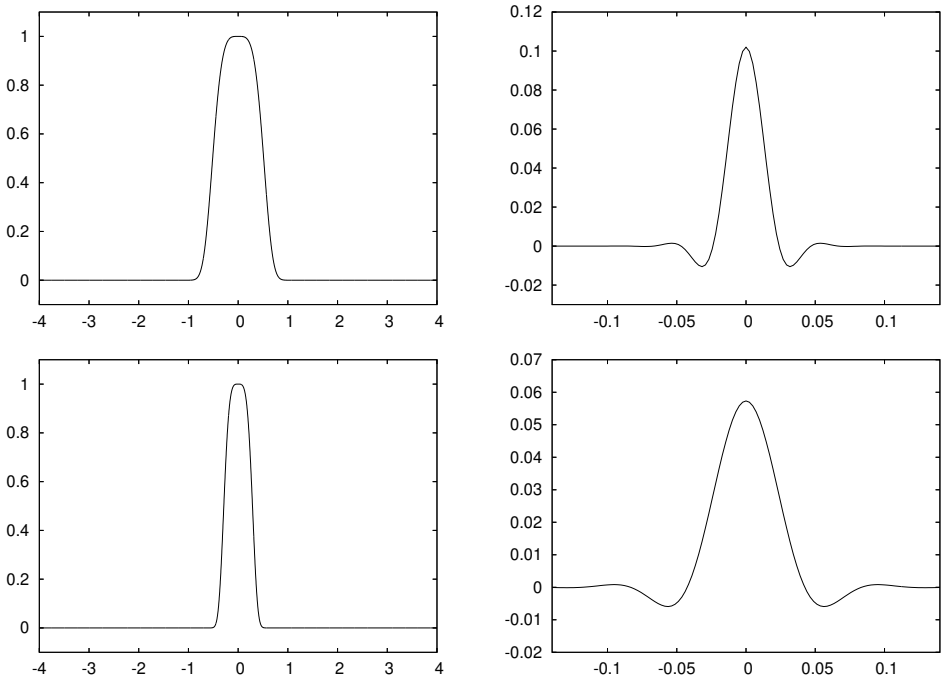


Figure 2.5: Behaviour of the multipliers and convolution kernels for fractional order diffusion with order $\alpha = 2$ and stopping time $t = 1$ for different λ. *Left:* Fourier multipliers G_λ^α. *Right:* Corresponding convolution kernels p_λ^α. *Top:* $\lambda = 10$. *Bottom:* $\lambda = 100$.

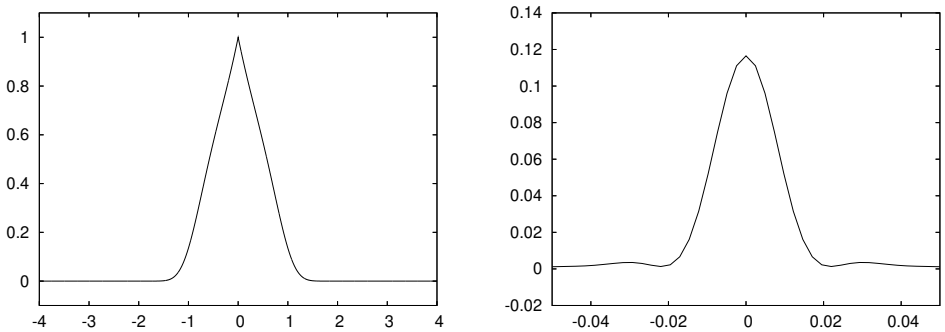


Figure 2.6: Positive combination of two kernel orders $\alpha_1 = 0.5$ and $\alpha_2 = 2$ with weights $\lambda_1 = 1$ and $\lambda_2 = 1$. *Left:* Fourier multiplier G_λ^α. *Right:* Corresponding convolution kernel p_λ^α.

We would like to mention that the convolution kernels $p_\lambda^\alpha(\cdot,t)$ were already discussed by Iijima in [102] and [103] for $\alpha \in \mathbb{N}$. With this definition we are able to express the Fourier backtransform of the solution of (2.17) as convolution:

$$u(x,t) \;=\; \left(p_{\lambda_m}^{\alpha_m}(\cdot,t) * \ldots * p_{\lambda_1}^{\alpha_1}(\cdot,t) * f\right)(x) \;\; . \tag{2.19}$$

It is an interesting feature of (2.18) and (2.19) that one can successively add different derivative orders to the right-hand side of (2.17) and obtain the particular solution step by step by convolution with corresponding kernels. The Figures 2.4, 2.5, and 2.6 show some examples for the Fourier multipliers and the corresponding convolution kernels coming from diffusion with fractional orders. Similar to the examples for regularisation, we first take a look at the behaviour of kernels with one single order when this order changes (see Figure 2.4). Again we notice that for orders $\alpha \leq 1$, we obtain positive convolution kernels. In the case of diffusion, orders $\alpha > 1$ lead to more oscillations of the convolution kernels than in the regularisation case. For larger λ the kernels show a similar behaviour and get wider, as it can be seen in Figure 2.5. As the example in Figure 2.6 indicates, it is also possible for fractional order diffusion to construct positive combinations of low and high orders.

As in the last section for regularisation, we also express fractional order diffusion as linear operator.

Definition 2.7 (Fractional Order Diffusion)
We choose fractional derivative orders $\alpha_1, \ldots, \alpha_m > 0$ and the corresponding weights $\lambda_1, \ldots, \lambda_m > 0$. For every $t \geq 0$ we define the linear filtering operator $\mathcal{T}_t : \mathscr{L}^2(\mathbb{R}^n) \longrightarrow \mathscr{L}^2(\mathbb{R}^n)$ as

$$\mathcal{T}_t f \;:=\; \mathcal{F}^{-1}\mathcal{M}_{G_{\lambda_m}^{\alpha_m}(\cdot,t)} \cdot \ldots \cdot \mathcal{M}_{G_{\lambda_1}^{\alpha_1}(\cdot,t)}\mathcal{F} f \;\; . \tag{2.20}$$

With respect to stability and smoothness of the solutions, we see that these diffusion operators have very convenient properties:

Proposition 2.8 (Stability and Regularity of Diffusion)

1. *For all $t \geq 0$ the operator $\mathcal{T}_t$ is continuous with respect to the norm in $\mathscr{L}^2(\mathbb{R}^n)$ with $\|\mathcal{T}_t\| \leq 1$.*
2. *For positive real filter orders $\alpha_1, \ldots, \alpha_m > 0$ we have $\mathcal{T}_t f \in \mathscr{C}^\infty(\mathbb{R}^n)$ for initial data $f \in \mathscr{L}^2(\mathbb{R}^n)$.*

Proof:

1. We have $0 \leq G_\lambda^\alpha(\xi,t) \leq 1$ for all $t \geq 0$, $\alpha, \lambda > 0$ and all $\xi \in \mathbb{R}^n$. An upper bound for the norm of $\mathcal{T}_t$ is given by

$$\|\mathcal{T}_t\| \;\leq\; \left\|\mathcal{F}^{-1}\right\| \left(\prod_{k=1}^{m} \left\|\mathcal{M}_{G_{\lambda_k}^{\alpha_k}(\cdot,t)}\right\|\right) \|\mathcal{F}\| \;\leq\; 1$$

with Lemma 2.1 and the fact that $\mathcal{F}$ is $\mathscr{L}^2$-isometric.

2. We note that
$$\lim_{|x|\longrightarrow\infty} x^\gamma \exp(-|x|^{2\alpha}) \;=\; 0$$
for all $\gamma \in \mathbb{N}^n$ where the power is defined componentwise. Thus we have
$$\xi^\gamma \exp\left(-t\sum_{k=1}^{m}\lambda_k|\xi|^{2\alpha_k}\right) \;\in\; \mathscr{L}^\infty(\mathbb{R}^n)\ .$$
Let $\gamma \in \mathbb{N}^n$ be an arbitrary derivative with order $|\gamma|$. The Fourier transform of the corresponding weak derivative of our filtered image
$$\mathcal{F}\mathcal{D}^\gamma\left(\mathcal{T}_t f\right) \;=\; i^{|\gamma|}\xi^\gamma \exp\left(-t\sum_{k=1}^{m}\lambda_k|\xi|^{2\alpha_k}\right)\hat{f} \tag{2.21}$$
is in $\mathscr{L}^2(\mathbb{R}^n)$ as the product of $\hat{f}$ with a bounded function. Since this works for all derivatives of order $|\gamma|$, we have shown that $\mathcal{T}_t f \in \mathscr{H}^{|\gamma|}(\mathbb{R}^n)$. Since $|\gamma|$ was arbitrary we know with the Sobolev embedding theorem (see [182, Chapter 4, Corollary 1.4]) that for each $m \in \mathbb{N}$ there is an $u \in \mathscr{C}^m(\mathbb{R}^n)$ with $u = \mathcal{T}_t f$ almost everywhere. In that sense the filtering results for fractional order diffusion processes are arbitrary smooth for all stopping times $t > 0$.

□

We have seen that in opposition to regularisation, the diffusion processes have much stronger smoothing behaviour. Furthermore, linear diffusion filtering satisfies a number of scale-space properties:

Proposition 2.9 (Properties of Diffusion)

1. *The set of linear diffusion operators* $\{\mathcal{T}_t : t \geq 0\}$ *is a semigroup. We have* $\mathcal{T}_0 = \mathcal{I}$ *and* $\mathcal{T}_{t_1}\mathcal{T}_{t_2} = \mathcal{T}_{t_1+t_2}$ *for all* $t_1, t_2 \geq 0$.
2. *For all* $t \geq 0$ *the average grey value is invariant under* $\mathcal{T}_t$.
3. *The continuous filtering operator is invariant under translations and rotations.*

Proof:

1. Since $G_\lambda^\alpha(\cdot, 0) = \exp(0) = 1$ it is clear that $\mathcal{T}_0 = \mathcal{I}$. For $t_1, t_2 > 0$ and $\xi \in \mathbb{R}$ one can directly verify $G_\lambda^\alpha(\xi, t_1)G_\lambda^\alpha(\xi, t_2) = G_\lambda^\alpha(\xi, t_1 + t_2)$. In the case of a single order α we have with the second statement of Lemma 2.1:
$$\begin{aligned}\mathcal{T}_{t_1}\mathcal{T}_{t_2} &= \mathcal{F}^{-1}\mathcal{M}_{G_\lambda^\alpha(\cdot,t_1)}\mathcal{F}\mathcal{F}^{-1}\mathcal{M}_{G_\lambda^\alpha(\cdot,t_2)}\mathcal{F}\\ &= \mathcal{F}^{-1}\mathcal{M}_{G_\lambda^\alpha(\cdot,t_1+t_2)}\mathcal{F}\\ &= \mathcal{T}_{t_1+t_2}\ .\end{aligned}$$
The same proof also works for multiple filter orders.

2. Average grey value invariance is guaranteed by $G_\lambda^\alpha(0,t) = 1$ for all $t, \lambda \geq 0$ and $\alpha > 0$.

3. Translational invariance follows directly from the representation of the operator $\mathcal{T}_t$ as convolution with kernels of the type p_λ^α as in (2.19).

   The multipliers G_λ^α only depend on the quadratic distance $|\xi|^2$ to the origin and thus are rotationally symmetric. This implies rotational invariance of diffusion. □

Scale invariance is not given in the framework considered above: To achieve this property we have to restrict ourselves to a single derivative order and one spatial dimension.

Proposition 2.10 (Scale Invariance of Diffusion)
In one spatial dimension ($n = 1$), the diffusion filter $\mathcal{T}_f := \mathcal{F}^{-1}\mathcal{M}_{G_\lambda^\alpha(\cdot,t)}\mathcal{F}$ with only one single derivative order is scale invariant in the following sense: For every $\sigma > 0$ and every $t > 0$ there is a $\tilde{t} > 0$ such that

$$\left(\mathcal{T}_t f\left(\frac{\cdot}{\sigma}\right)\right)(x) = \left(\mathcal{T}_{\tilde{t}} f(\cdot)\right)\left(\frac{x}{\sigma}\right) \quad .$$

Linear filtering with a combination of several orders or in multiple dimensions is not scale invariant in this sense.

Proof: It can be shown by elementary calculations that $\tilde{t} = \frac{t}{\sigma^{2\alpha}}$ is the unique value satisfying the above condition. Since $\tilde{t}$ depends on the order α such a time can not exist for a combination of different orders. □

Remark 2.11 (Extension to Multi-Channel Images)
We have seen that regularisation and diffusion both can be described as convolutions, and the convolution kernels are independent from the initial data. Thus the reasoning given above would also work for multi-channel images $u : \mathbb{R}^n \longrightarrow \mathbb{R}^m$. As the result would be to treat all channels independently, we only have described the case $m = 1$ of one channel here.

2.1.4 Semi-Discrete Linear Filtering

For practical purposes a space-discrete formulation of generalised linear filtering can be very useful. In this section we give a matrix representation for the filters which can be understood as a finite-dimensional analogue of the operators given above. Again we consider the n-dimensional case with an equidistant orthogonal grid with N^n pixels x_j for even N, indexed by $j \in J := \{0, \ldots, N-1\}^n$. The index of the central pixel in this rectangular grid is given by $c := \frac{1}{2}(N, \ldots, N)^T$.

In correspondence to the operator $\mathcal{F}$ we define the discrete Fourier transform matrix $F \in \mathbb{C}^{(N^n)\times(N^n)}$ by

$$F := \frac{1}{\sqrt{N^n}} \left(\exp\left(-\frac{2\pi i \langle j-c, k\rangle}{N^n} \right) \right)_{j,k\in J} . \tag{2.22}$$

Since the rows of F are orthonormal in $\mathbb{C}^{(N^n)}$, F is unitary and its inverse is given by its transposed complex conjugated matrix $\overline{F}^T$. The matrix-vector product of F with $g \in \mathbb{R}^{(N^n)}$ yields the Fourier coefficients $Fg =: \hat{g} \in \mathbb{C}^{(N^n)}$. We define the analogue to the multiplication operator $\mathcal{M}$ as the diagonal matrix

$$M_f := \operatorname{diag}\left(f\left(\frac{2\pi\,(j-c)}{N^n} \right) \right)_{j\in J} \tag{2.23}$$

which multiplies a vector with the values of a function $f : [-\pi,\pi)^n \longrightarrow \mathbb{C}$ at the equidistant grid points in the Fourier domain.

Definition 2.12 (Semi-Discrete Regularisation and Diffusion Matrices) *As space-discrete analogues to (2.12) and (2.20), for $\alpha, \beta, \lambda \in \mathbb{R}^m_+$ and $t > 0$ we define the regularisation matrix $R^\alpha_\beta := \overline{F}^T M_{r^\alpha_\beta} F$ and the linear diffusion matrix via $T_t := \overline{F}^T M_{G^{\alpha_m}_{\lambda_m}(\cdot,t)} \cdot \ldots \cdot M_{G^{\alpha_1}_{\lambda_1}(\cdot,t)} F$.*

In the discrete setting, the corresponding filtering techniques are thus defined as diagonalised matrices where the multiplier functions directly yield the eigenvalues. This simplifies the consideration of the scale-space properties in the following:

Proposition 2.13 (Properties of Regularisation)

1. *Semi-discrete regularisation is linear.*
2. *The average grey value is invariant under the operators R^α_β for all nonnegative regularisation weights. For $\beta \to \infty$ in all components the solution converges towards the average grey value, i. e.*

$$\lim_{\beta\to\infty} R^\alpha_\beta f \;=\; (\mu,\ldots,\mu)^T \quad \textit{with } \mu \;:=\; \frac{1}{N^n}\sum_{j\in J} f_j \;.$$

Proof: The scaled average grey value can be written as $\hat{f}_0 = \frac{1}{\sqrt{N^n}} \sum_{j\in J} f_j$. This coefficient is left unchanged by the diagonal matrices $M_{r^\alpha_\beta}$ since $r^\alpha_\beta(0) = 1$. Thus claimed convergence follows from $\lim_{\beta\to\infty} r^\alpha_\beta(\xi) = 0$ for all $\xi \neq 0$. □

We see that diffusion has the same scale-space properties, and in addition, the diffusion operators form a semigroup:

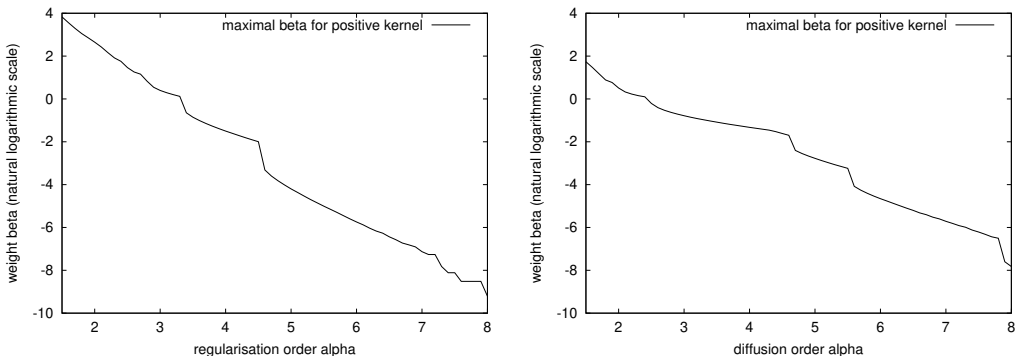


Figure 2.7: Positive combinations of derivatives of order 0.5 with higher orders, $n = 1$. *Left:* Regularisation. *Right:* Diffusion filtering.

Proposition 2.14 (Properties of Diffusion)

1. *Semi-discrete diffusion is linear.*
2. *The set of operators* $\{T_t : t \geq 0\}$ *is a semigroup.*
3. *The average grey value is invariant under the operators* T_t *for all* $t \geq 0$*, and we have convergence towards the average grey value for* $t \longrightarrow \infty$*.*

Proof: The proof of the second statement is analogous to the proof of Proposition 2.9 exchanging the operators $\mathcal{F}$ and $\mathcal{M}$ by their finite-dimensional counterparts F and M. The third statement is proven as in the regularisation case. □

So far, we have not considered invariance with respect to translation and rotation in the discrete setting. As rotations can only be approximated on a rectangular pixel grid, we only obtain approximative rotational invariance here. On the other hand, translational invariance is possible with one restriction: We have to consider the finite extension of our pixel grid and take a look at what happens at the boundaries. In our setting here, the filters are translationally invariant if the translations are cyclic shifts.

2.1.5 Numerical Experiments

In the first numerical experiment we investigate the possibility of building linear combinations with different derivative orders such that the regularisation and diffusion filters satisfy a maximum-minimum property. Knowing from Subsection 2.1.3 that combinations of two orders are no longer scale-invariant we try to preserve one scale-space property at the expense of the other. To reduce the number of possible combinations we consider diffusion equations of the form

$$\partial_t u \;=\; -\left(\sqrt{-\Delta} + \beta\,(-\Delta)^{\alpha}\right) u \tag{2.24}$$

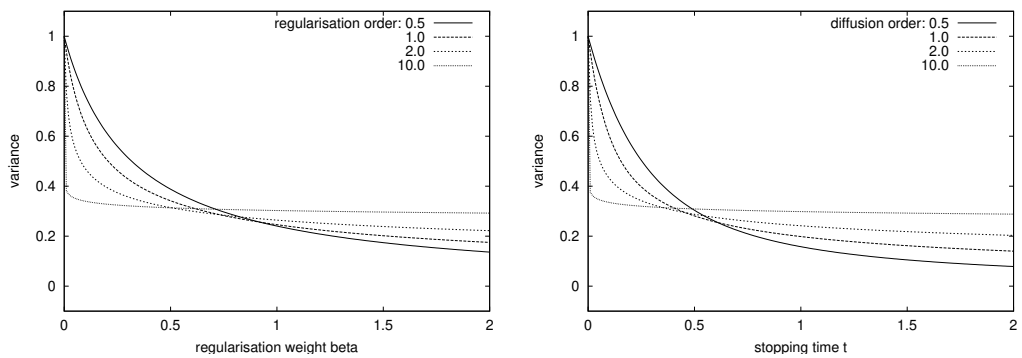


Figure 2.8: Variance diminishing properties of fractional order regularisation and diffusion filtering, $n = 1$. Variance depending on regularisaton weight/diffusion stopping time. *Left:* Regularisation. *Right:* Diffusion filtering.

and the corresponding regularisation. For α between 1.5 and 8, we started with $\beta = 0$ and increased it as long as nonnegative convolution kernels were obtained. The maximal values of β are shown in Figure 2.7. This experiment shows the usefulness of the Poisson scale-space: Using a Gaussian scale-space instead makes it impossible to find a weight $\beta \neq 0$ that leads to a nonnegative combination. In that sense the fractional order scale-space has a clear advantage in comparison with the integer order ones.

In our second experiment we study the variance diminishing properties of different filters $\mathcal{R}$ and $\mathcal{T}$. Figure 2.8 shows the variance of a white noise signal depending on regularisation weight / stopping time. We visualise the parameters needed for reducing the variance of a white noise and a step edge signal to half of its value in Figure 2.9. The experiments show a similar behaviour of regularisation and corresponding diffusion techniques in terms of variance reduction. We note that higher orders lead to the same variance decay with smaller stopping times.

2.2 Nonlinear Filtering with Higher Derivatives

After investigating linear filtering methods which could be expressed as convolutions, we now turn our attention to nonlinear methods. Since we start with a variational formulation for the nonlinear filters, it is necessary to provide some technical background in terms of the calculus of variations in Subsection 2.2.1. This knowledge will be immediately applied in the following Subsection 2.2.2 to characterise possible minimisers for a certain class of nonlinear higher order regularisation functionals. The necessary conditions for such minimisers are PDEs which directly lead us to higher order nonlinear diffusion equations which are the central filter class discribed in this section. A property of these diffusion equations which is especially interesting in practice, namely local feature enhancement, will

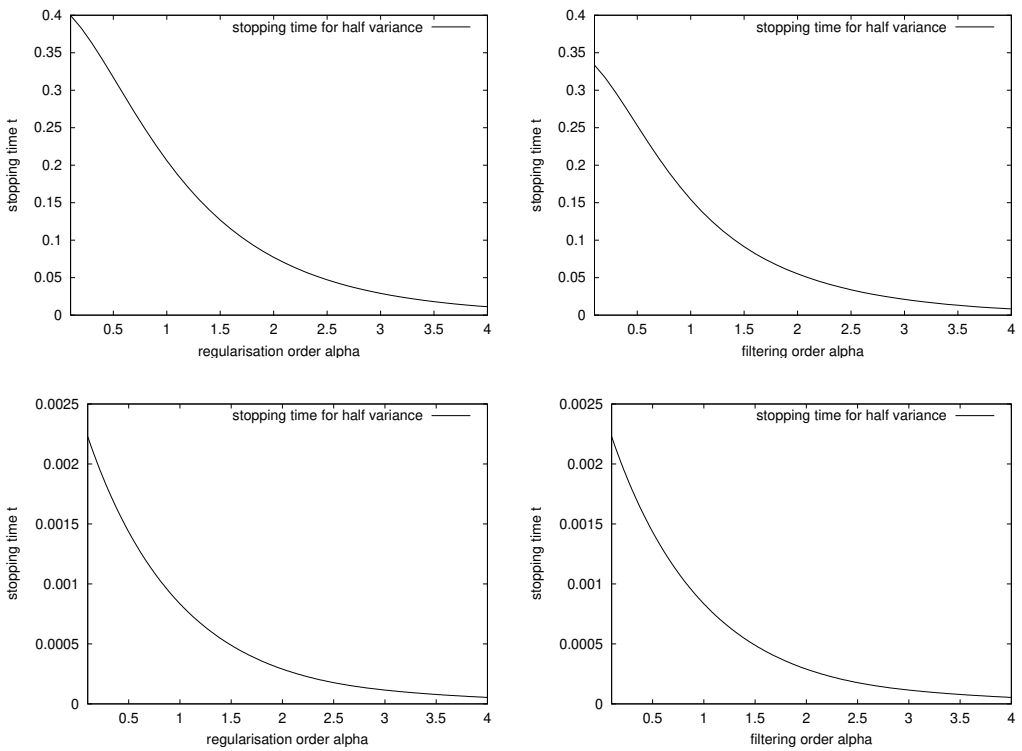


Figure 2.9: Regularisation weight/diffusion stopping time for reducing the variance to half its value, $n = 1$. *Left column:* Regularisation. *Right column:* Diffusion filtering. *Top row:* Experiment for white noise signal. *Bottom row:* Experiment for step edge signal.

be enlightened in Subsection 2.2.3. To visualise these properties in practice, we will formulate possible discretisations in Section 2.2.4 and give reasons for similar behaviour as in the continuous setting. Some numerical results are displayed in Subsection 2.2.5 showing the properties of fourth order filtering and comparing the denoising quality with second order filters.

2.2.1 Calculus of Variations

First we determine necessary conditions for general variational problems involving higher derivatives. Of special importance for us will be the natural boundary conditions, since they will allow to prove some important properties of image filtering methods later on.

Let $\Omega \subset \mathbb{R}^n$ be an open set such that the boundary $\partial\Omega$ is piecewise smooth and $x = (x_1, \ldots, x_n)^T \in \Omega$. We start with a general variational functional

$$\mathscr{E}(u) \;:=\; \int_\Omega E(x, u, \mathcal{D}u, \ldots, \mathcal{D}^p u)\; dx \tag{2.25}$$

which depends on a function $u : \Omega \longrightarrow \mathbb{R}^m$ and its partial derivatives up to the order $p \in \mathbb{N}$. In the following, we will shortly write $[x, u, p] := (x, u, \mathcal{D}u, \ldots, \mathcal{D}^p u)$ for the arguments of the integrand E. Further, $u^i : \Omega \longrightarrow \mathbb{R}$ for $i \in \{1, \ldots, m\}$ stands for the i-th component of u. Let $V := \{x_1, \ldots, x_n\}$ be the set of variables and $\beta = (\beta_1, \ldots, \beta_p) \in V^p$ a vector of p components in this set. For a function u the partial derivative will be denoted by

$$\mathcal{D}^\beta u \;:=\; \partial_{\beta_m} \cdot \ldots \cdot \partial_{\beta_1} u \;\; .$$

The derivative order is then $|\beta| := p$. We explicitly do not use multiindices here, since we have to take care of the order of the derivatives for the boundary conditions. The revearsal of β will be denoted by $\tilde{\beta} := (\beta_p, \ldots, \beta_1)$.

Having these notations at hand, we can formulate necessary conditions for a minimiser of $\mathscr{E}$:

Proposition 2.15 (Euler-Lagrange Equations)
A minimiser of the energy functional (2.25) necessarily satisfies the so-called Euler-Lagrange equations

$$\sum_{|\beta| \leq p} (-1)^{|\beta|} \mathcal{D}^{\tilde{\beta}} E_{\mathcal{D}^\beta u^i}([x, u, p]) \;=\; 0 \quad \textit{for } i = \{1, \ldots, m\} \;\; . \tag{2.26}$$

Proof: To determine a minimiser of the function $\mathscr{E}$, we are interested in the behaviour of $\mathscr{E}$ under perturbations of its argument. This is done by choosing an arbitrary perturbation function $\eta : \Omega \longrightarrow \mathbb{R}^m$ and considering the function $\Psi(\varepsilon) := \mathscr{E}(u + \varepsilon\eta)$. It is a classical procedure in the calculus of variations [88, 86] to use the necessary condition $\Psi'(0) = 0$ to characterise minimising functions u. The derivative of Ψ is

$$\Psi'(\varepsilon) \;=\; \sum_{|\beta| \leq p} \sum_{i=1}^{m} \int_\Omega E_{\mathcal{D}^\beta u^i}([x, u + \varepsilon\eta, p]) \mathcal{D}^\beta \eta^i \; dx \;\; . \tag{2.27}$$

At the point zero, we obtain

$$\begin{aligned}
\Psi'(0) \;&=\; \sum_{|\beta| \leq p} \sum_{i=1}^{m} \int_\Omega E_{\mathcal{D}^\beta u^i}([x, u, p]) \mathcal{D}^\beta \eta^i \; dx \\
&=\; \sum_{|\beta| \leq p} (-1)^{|\beta|} \sum_{i=1}^{m} \int_\Omega \left(\mathcal{D}^{\tilde{\beta}} E_{\mathcal{D}^\beta u}([x, u, p]) \right) \eta^i \; dx \qquad (2.28) \\
&\quad + \sum_{|\beta| \leq p} \sum_{k=1}^{|\beta|} (-1)^{|\beta|-k} \sum_{i=1}^{m} \int_{\partial\Omega} \left(\partial_{\beta_{k+1}} \ldots \partial_{\beta_{|\beta|}} E_{\mathcal{D}^\beta u^i} \right) \left(\partial_{\beta_{k-1}} \ldots \partial_{\beta_1} \eta^i \right) \nu_{\beta_k} dx.
\end{aligned}$$

The outer normal of the boundary $\partial\Omega$ is denoted with ν here. We have omitted the set of arguments $[x, u, p]$ in the last integral for simplicity. Under the

assumption that $\eta \in \mathscr{C}_C(\Omega)$ has compact support the boundary terms vanish:

$$\int_\Omega \Big(\sum_{|\beta|\leq m} (-1)^{|\beta|} \sum_{i=1}^{m} \mathcal{D}^{\tilde{\beta}} E_{\mathcal{D}^\beta u^i}([x,u,p]) \Big)\, \eta^i \; dx \;=\; 0 \tag{2.29}$$

for $i \in \{1,\dots,m\}$. Since this holds for all $\eta \in \mathscr{C}_C(\Omega)$, the fundamental lemma of the calculus of variations [88] can be applied. We obtain the set of m Euler-Lagrange equations

$$\sum_{|\beta|\leq p} (-1)^{|\beta|} \mathcal{D}^{\tilde{\beta}} E_{\mathcal{D}^\beta u^i}([x,u,p]) \;=\; 0 \quad \text{for } i = \{1,\dots,m\} \tag{2.30}$$

as a necessary condition for a minimiser u of $\mathscr{E}$. □

In this proof we have restricted the test functions η to the interior of Ω. Without this restriction, the boundary terms in (2.28) do not vanish. In our applications, Ω is usually a rectangular domain. That means we can find test functions where the partial derivatives vanish independently at the boundary. This gives us the following conditions on u:

Proposition 2.16 (Natural Boundary Conditions)
If we do not impose other restrictions at the boundary, the minimiser u naturally satisfies the following property: For all $i \in \{1,\dots,m\}$, all $k \in \{1,\dots,p\}$, and all $\gamma \in V^{k-1}$

$$\sum_{\substack{k\leq|\beta|\leq p \\ (\beta_1,\dots,\beta_{k-1})=\gamma}} (-1)^{|\beta|-k} \Big(\partial_{\beta_{k+1}} \dots \partial_{\beta_{|\beta|}} E_{\mathcal{D}^\beta u^i} \Big)\, \nu_{\beta_k} \;=\; 0 \tag{2.31}$$

on $\partial\Omega$.

With these results we are able to consider variational methods with higher derivative orders for multi-channel images in n spatial dimensions. In the next section, we will turn our attention to a very special class of representatives of these methods which comprises many classical approaches.

2.2.2 Higher Order Regularisation and Diffusion

For higher order nonlinear regularisation we are searching for a minimiser of the energy functional

$$\mathscr{E}(u) \;=\; \int_\Omega \Big(\sum_{i=1}^{m} (u^i - f^i)^2 \;+\; \alpha\, \Psi\Big(\sum_{i=1}^{m} \sum_{|\beta|=p} |\mathcal{D}^\beta u^i|^2 \Big) \Big)\; dx \tag{2.32}$$

for $\alpha > 0$. The regulariser depends here on the sum of the squared derivatives of order p. This can be motivated by the fact that the smoothness term vanishes if u is a polynomial of degree $p-1$. Special cases of this functional for $p = 2$ are

considered by Lysaker et al. [122], for example. With Proposition 2.15 we obtain the following Euler-Lagrange equations:

$$\frac{u^i - f^i}{\alpha} \;=\; (-1)^{p+1} \sum_{|\beta|=p} \mathcal{D}^{\tilde{\beta}} \Big(\Psi' \Big(\sum_{j=1}^{m} \sum_{|\gamma|=p} |\mathcal{D}^\gamma u^j|^2 \Big) \mathcal{D}^\beta u^i \Big) \tag{2.33}$$

for all $i \in \{1, \ldots, m\}$ as necessary condition for this minimiser. The corresponding natural boundary conditions are given by Proposition 2.16: For all $i \in \{1, \ldots, m\}$, all $k \in \{1, \ldots, p\}$, and all $\gamma \in V^{k-1}$ we have

$$\sum_{\substack{|\beta|=p \\ (\beta_1, \ldots, \beta_{k-1})=\gamma}} (-1)^{p-k} \left(\partial_{\beta_{k+1}} \ldots \partial_{\beta_p} E_{\mathcal{D}^\beta u^i} \right) \nu_{\beta_k} \;=\; 0 \;. \tag{2.34}$$

It is possible to use these Euler-Lagrange equations directly to determine possible minimisers for the energy functional. Let us now turn our attention to some properties of these minimisers. Preservation of the average grey value is a basic property of classical regularisation techniques as discussed in Subsection 1.2.2. We will see that there is even a more general property for higher regularisation orders:

Proposition 2.17 (Moment Preservation of Regularisation)
All moments up to the order $p-1$ of all channels u^i of the regularisation solution do not change with $\alpha > 0$.

Proof: Choose a monomial $m(z) := x_1^{l_1} \cdot \ldots \cdot x_n^{l_n}$ of degree smaller than p, i. e. $\sum_{k=1}^n l_k < p$. Then we have $\mathcal{D}^\beta m = 0$ for all $|\beta| = p$. We multiply the Euler-Lagrange equation (2.33) and integrate to obtain

$$\begin{aligned} \int_\Omega m(z) u^i(z) \, dx \;&=\; \int_\Omega m(z) f^i(z) \, dx \\ &\quad + \alpha \int_\Omega m(z) (-1)^{p+1} \sum_{|\beta|=p} \mathcal{D}^{\tilde{\beta}} \Big(\Psi' \Big(\sum_{j=1}^{m} \sum_{|\gamma|=p} |\mathcal{D}^\gamma u^j|^2 \Big) \mathcal{D}^\beta u^i \Big) \, dx \;. \end{aligned}$$

We can calculate the second integral on the right-hand side and use to natural boundary conditions (2.34) to see that

$$\begin{aligned} &\int_\Omega m(z) (-1)^{p+1} \sum_{|\beta|=p} \mathcal{D}^{\tilde{\beta}} \Big(\Psi' \Big(\sum_{j=1}^{m} \sum_{|\gamma|=p} |\mathcal{D}^\gamma u^j|^2 \Big) \mathcal{D}^\beta u^i \Big) \, dx \\ &= \; (-1)^{p+1} \sum_{|\beta|=p} \underbrace{(\mathcal{D}^\beta m)}_{=0} \Psi' \Big(\sum_{j=1}^{m} \sum_{|\gamma|=p} |\mathcal{D}^\gamma u^j|^2 \Big) \mathcal{D}^\beta u^i \, dx \\ &\quad + (-1)^{p+1} \sum_{|\beta|=p} \sum_{k=1}^{p} (-1)^{p-k} \int_{\partial\Omega} \Big(\big(\partial_{\beta_{k+1}} \ldots \partial_{\beta_p} m \big) \cdot \\ &\qquad\qquad \cdot \left(\partial_{\beta_{k-1}} \ldots \partial_{\beta_1} \Psi' \left(\ldots \right) \mathcal{D}^\beta u^i \right) \nu_{\beta_k} \Big) \, dx \\ &= \; 0 \;. \end{aligned}$$

We have abbreviated the last term by omitting the argument of Ψ'. Together, this means that the moments up to order $p-1$ remain constant. □

In the following, we will not directly use Euler-Lagrange equations for data filtering, but we rather prefer to follow the reasoning already described in Subsection 1.2.2 to relate it to generalised diffusion equations. Again, one can understand the left-hand side of (2.33) as finite difference. Introducing an artificial time variable t in our function u and setting $u(\cdot,0) := f$ makes it possible to see the left-hand side as discretisation of a time-derivative of u with step size α. The whole equation (2.33) is then an implicit time discretisation of a nonlinear diffusion equation.

This motivates to use nonlinear higher order diffusion equations of the form

$$\begin{aligned} u^i(\cdot,0) &= f^i \\ \partial_t u^i &= (-1)^{p+1} \sum_{|\beta|=p} \mathcal{D}^{\tilde{\beta}} \Big(\Psi' \Big(\sum_{j=1}^{m} \sum_{|\gamma|=p} |\mathcal{D}^\gamma u^j|^2 \Big) \mathcal{D}^\beta u^i \Big) \end{aligned} \tag{2.35}$$

for $i \in \{1,\dots,m\}$ with the boundary conditions given in (2.34) for data filtering.

If we are not interested in the variational function and tend to consider the diffusion equation on its own, we will often write $g = \Psi'$ for the diffusivity function. Let us consider some special cases for (2.35) which will be relevant for us later on. In all of the following examples, we set the initial condition $u(\cdot,0) = f$ without explicitly stating it:

Example 2.18 (General Filter of Order 2p in One Dimension)
In one spatial dimension we consider filtering on an interval $(a,b) \subset \mathbb{R}$. Nonlinear diffusion of order $2p$ then denotes the equation

$$\partial_t u \;=\; (-1)^{p+1} \, \partial_x^p \Big(g\left((\partial_x^p u)^2 \right) \partial_x^p u \Big) \;. \tag{2.36}$$

The corresponding natural boundary conditions are in this case

$$\partial_x^k \Big(g\left((\partial_x^p u)^2 \right) \partial_x^p u \Big) \;=\; 0 \quad \text{for } k \in \{0,\dots,p-1\} \tag{2.37}$$

for $x \in \{a,b\}$. There are p constraints at each boundary pixel as generalisation of the homogeneous Neumann boundary conditions which are well-known from the case $p=1$.

Example 2.19 (Classical Perona-Malik Filter)
In the case $p=1$ and in two dimensions, we obtain the classical Perona-Malik equation [154]:

$$\partial_t u \;=\; \operatorname{div}\Big(g(|\nabla u|^2) \nabla u \Big) \tag{2.38}$$

with homogeneous Neumann boundary conditions $\partial_\nu u = 0$ and its multi-channel representatives [87], respectively.

Example 2.20 (Fourth Order Filter in 2-D)
The corresponding filter for $p = 2$ and TV diffusivity has been considered by Lysaker et al. [122]. Here we allow for a more general diffusivity function g, and the corresponding filter looks like

$$\begin{aligned} \partial_t u &= -\partial_{xx}\Big(g(\|H(u)\|_F^2)\, u_{xx}\Big) - \partial_{yx}\Big(g(\|H(u)\|_F^2)\, u_{xy}\Big) \\ &\quad -\partial_{xy}\Big(g(\|H(u)\|_F^2)\, u_{yx}\Big) - \partial_{yy}\Big(g(\|H(u)\|_F^2)\, u_{yy}\Big)\ . \end{aligned} \tag{2.39}$$

The natural boundary conditions are in this case three equations:

$$\begin{aligned} g\, u_{xx}\nu_x + g u_{xy}\nu_y &= 0\ , \\ g\, u_{yx}\nu_x + g u_{yy}\nu_y &= 0\ , \\ (\partial_x(g\, u_{xx}) + \partial_y(g\, u_{xy}))\,\nu_x + (\partial_x(g\, u_{xy}) + \partial_y(g\, u_{yy}))\,\nu_y &= 0 \end{aligned}$$

on $\partial\Omega$. We have omitted the argument $\|H(u)\|_F^2$ of g for better readability.

Remark 2.21 (Presmoothing and Well-Posedness)
For the classical Perona-Malik filter, Catté et al. [32] have shown that introducing a mollifier in the argument of the diffusivity makes it possible to prove well-posedness. Similar methods have been applied by Greer and Bertozzi [92] to show existence and regularity of solutions for the models by Tumblin, Turk [189] and Wei [195]. Applied to our general model, the modified diffusion equation reads as

$$\partial_t u^i = (-1)^{p+1} \sum_{|\beta|=p} \mathcal{D}^{\tilde{\beta}}\Big(\Psi'\Big(\sum_{j=1}^{m}\sum_{|\gamma|=p} |\mathcal{D}^\gamma u_\sigma^j|^2\Big)\mathcal{D}^\beta u^i\Big) \tag{2.40}$$

where $u_\sigma^j := G_\sigma * u^j$ with an n-dimensional Gaussian kernel G_σ of standard deviation σ.

Later on we will see from the numerical examples that discretisations of (2.35) also work without such a regularisation. This indicates that discretisation has a regularising effect itself, as it has been proven for the classical Perona-Malik case by Weickert and Benhamouda [202]. We have seen in the previous section that instead of presmoothing with a Gaussian kernel, one could also use more general fractional diffusion equations. They all have the property to yield arbitrary smooth results and thus are suited for regularising the nonlinear diffusion equations.

After these remarks concerning well-posedness we consider some scale-space properties of nonlinear diffusion.

Proposition 2.22 ($\mathscr{L}^2$-Stability)
If a classical solution u of higher order nonlinear diffusion (2.35) exists which is continuously differentiable in the time variable t and $2p$ times continuously differentiable in the space variable, the $\mathscr{L}^2$-norm of all channels $u^i(\cdot,t)$ for all $i \in \{1,\dots,m\}$ is monotonically decreasing with $t \geq 0$.

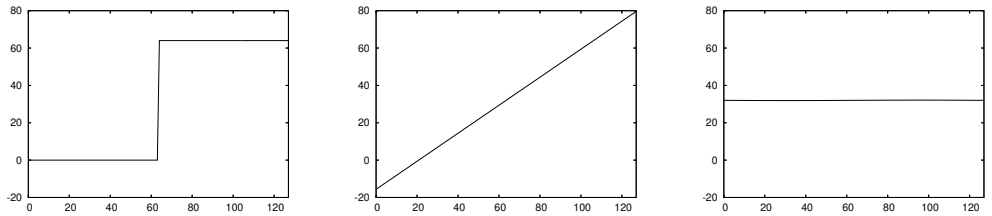


Figure 2.10: Stability of diffusion filtering and importance of boundary conditions. *Left:* Original signal, 128 pixels. *Middle:* Steady state of fourth order linear diffusion filtering with natural boundary conditions. *Right:* Same with periodic boundary conditions.

Proof: We calculate the time derivative of the $\mathscr{L}^2$-norm for one channel u^i of the image using partial integration and the boundary conditions (2.34):

$$
\begin{aligned}
\partial_t \left(\frac{1}{2} \int_\Omega |u^i|^2 \, dx \right) &= \int_\Omega u^i (\partial_t u^i) dx \\
&= (-1)^{p+1} \sum_{|\beta|=p} \int_\Omega u^i \mathcal{D}^{\tilde{\beta}} \Big(\Psi' \Big(\sum_{j=1}^{m} \sum_{|\gamma|=p} |\mathcal{D}^\gamma u^j|^2 \Big) \mathcal{D}^\beta u^i \Big) \, dx \\
&= (-1)^{p+1} \sum_{|\beta|=p} (-1)^p \int_\Omega (\mathcal{D}^\beta u^i)^2 \, \Psi' \Big(\sum_{j=1}^{m} \sum_{|\gamma|=p} |\mathcal{D}^\gamma u^j|^2 \Big) \, dx \\
&\quad + (-1)^{p+1} \sum_{|\beta|=p} \sum_{k=1}^{p} (-1)^{p-k} \int_{\partial\Omega} \Big(\big(\partial_{\beta_{k+1}} \dots \partial_{\beta_p} u \big) \cdot \\
&\qquad \cdot \big(\partial_{\beta_{k-1}} \dots \partial_{\beta_1} \Psi' \left(\dots \right) \mathcal{D}^\beta u^i \big) \, \nu_{\beta_k} \Big) \, dx \\
&= - \sum_{|\beta|=p} \int_\Omega \underbrace{(\mathcal{D}^\beta u^i)^2}_{\geq 0} \cdot \underbrace{\Psi' \Big(\sum_{j=1}^{m} \sum_{|\gamma|=p} |\mathcal{D}^\gamma u^j|^2 \Big)}_{\geq 0} \, dx \\
&\leq 0 \; .
\end{aligned}
$$

We see that for this property, it is essential that Ψ' is larger or equal to zero. Since Ψ is a penaliser function, this assumption is reasonable. Since we assumed that the solution is continuously differentiable in time, this shows that the $\mathscr{L}^2$-norm is monotonically decreasing. □

To illustrate this stability property, we show an example in Figure 2.10. It is visible that linear fourth order diffusion does not satisfy a maximum-minimum principle and thus is not stable in the $\mathscr{L}^\infty$-sense. This example also gives an additional motivation why the boundary conditions are so important for our considerations: Using periodic boundary conditions changes the behaviour of the

process significantly.

Besides the stability of the process, we are interested in characterisations of the simplification behaviour. For Perona-Malik filtering, it is well-known that the average grey value is left unchanged during the filtering process. Filtering with periodic boundary conditions has also led to the average grey value in the example shown in Figure 2.10. The filter with natural boundary conditions in this case yields a linear approximation to the initial signal and thus preserves more characteristic properties of the initial signal. The next proposition makes this precise for the continuous setting:

Proposition 2.23 (Moment Preservation for Diffusion)
Under the assumptions of Proposition 2.22, all moments up to the order $p-1$ of all channels u^i of the solution are constant in time $t \geq 0$.

Proof: Choose a monomial $m(z) := x_1^{l_1} \cdot \ldots \cdot x_n^{l_n}$ such that its degree is smaller than p, i. e. $\sum_{k=1}^n l_k < p$. Then we have $\mathcal{D}^\beta m = 0$ for all $|\beta| = p$. We use this to calculate the time-derivative of the corresponding moment:

$$\begin{aligned}
&\partial_t \left(\int_\Omega m(z) u^i dx \right) \\
&= \int_\Omega m(z) \partial_t u^i dx \\
&= (-1)^{p+1} \sum_{|\beta|=p} \int_\Omega m(z) \mathcal{D}^{\tilde{\beta}} \Big(\Psi' \Big(\sum_{j=1}^m \sum_{|\gamma|=p} |\mathcal{D}^\gamma u^j|^2 \Big) \mathcal{D}^\beta u^i \Big) \, dx \\
&= 0
\end{aligned}$$

as we have already calculated in the proof of Proposition 2.17. □

In the discrete setting, we are going to give an analogue to this statement in Proposition 2.25. Before turning our attention to the discrete setting, we are going to describe some aspects of nonlinear higher order filters that are crucial for the quality of the results: the possibility of not only preserving, but also enhancing features of the data like edges.

2.2.3 Local Feature Enhancement

Though classical nonlinear diffusion simplifies signals or images, it may also enhance important local features such as edges. This section discusses higher order diffusion from this point of view. We are going to work only with equations in the one-dimensional case in this subsection. As it can be seen with the numerical experiments later on, this does not mean that the results are restricted to 1-D. Even in higher dimensions, an image locally can be decomposed in 1-D feature directions: Let an edge in a 2-D image serve as example. Locally one can decompose the image in the two directions along the edge where the grey value does

not change, and across the edge where the change it maximal. This clearly also works with other low-dimensional features such as ridges. It is the same concept which we also meet in the filters respecting local feature directions considered in Chapter 4. In that sense one can use the reasoning given in this subsection to explain the behaviour of filters in multiple dimensions, too.

Second Order Filtering and Edge Enhancement

We have already shortly sketched this case in Subsection 1.2.2. To determine the possibility of edge enhancement for special diffusivities g one usually considers the the flux function $\Phi(s) := g(s^2)s$. One can rewrite the one-dimensional second-order nonlinear diffusion equation (2.36) for $p = 1$ yielding

$$\begin{aligned} \partial_t u &= \partial_x\Big(\Phi(\partial_x u)\Big) \\ &= \Phi'\left(\partial_x u\right)\partial_x^2 u \\ &= \left(2g'\left(\left(\partial_x u\right)^2\right)\left(\partial_x u\right)^2 + g\left(\left(\partial_x u\right)^2\right)\right)\partial_x^2 u \ . \end{aligned}$$

In regions where $\Phi'(\partial_x u) > 0$ this equation behaves like a forward diffusion equation while in regions with $\Phi'(\partial_x u) < 0$ there is backward diffusion possible. In these regions with backward diffusion, an edge enhancing behaviour is plausible and can also be observed in practice [154].

Fourth Order Filtering

Now we take a closer look at the fourth order diffusion equation, i. e. we set $p = 2$ in (2.36) yielding

$$\partial_t u \ = \ -\partial_x^2\left(g\left(\left(\partial_x^2 u\right)^2\right)\partial_x^2 u\right) \ .$$

We expand the right-hand side of this equation and rewrite it as

$$\partial_t u \ = \ -\Big(2\left(\partial_x^3 u\right)^2\Phi_1\left(\partial_x^2 u\right)\Big)\partial_x^2 u - \Phi_2\left(\partial_x^2 u\right)\partial_x^4 u \tag{2.41}$$

using

$$\Phi_1(s) \ := \ 2g''(s^2)s^2 + 3g'(s^2) \quad \text{and} \quad \Phi_2(s) \ := \ 2g'(s^2)s^2 + g(s^2) \ .$$

Analogue to the second order case our argumentation is that (2.41) locally behaves similar to the linear equation

$$\partial_t u \ = \ -a\,\partial_x^2 u - b\,\partial_x^4 u$$

if the signs of the factors a and b are equal to the signs of Φ_1 and Φ_2. For $\Phi_1(\partial_x^2 u) < 0$ we expect some second order forward diffusion influence on the solution, whereas $\Phi_1(\partial_x^2 u) > 0$ leads to second order backward diffusion. Vice

versa, $\Phi_2(\partial_x^2 u) > 0$ ensures fourth order forward diffusion, and $\Phi_2(\partial_x^2 u) < 0$ fourth order backward diffusion.

It should be mentioned that Φ_2 always coincides with the function Φ in the second order case presented in Subsection 2.2.3. Also for orders higher than four, the sign of this function determines the diffusion property (forward or backward) of the highest order term which implies a certain similarity in the behaviour of several filtering orders. The main difference is the argument: Φ depends on the p-th derivative for $2p$-th order filtering.

Application to Commonly Used Diffusivities

After showing the general approach for fourth order diffusion in the last section we now apply it to several diffusivities commonly used in practice to describe their characteristic behaviour. In the following the diffusivities are ordered according to their forward-backward diffusion properties:

- **Forward Diffusion:**
  The diffusivity related to the regularisation approach by Charbonnier et al. [39] is given by $g(s^2) = (1 + s^2/\lambda^2)^{-\frac{1}{2}}$ and is known to perform forward diffusion in the second order case. By computing

  $$\Phi_1(s) \;=\; -\tfrac{3}{2\lambda^2}\left(1+\tfrac{s^2}{\lambda^2}\right)^{-\frac{5}{2}} < 0 \quad \text{and} \quad \Phi_2(s) \;=\; \left(1+\tfrac{s^2}{\lambda^2}\right)^{-\frac{3}{2}} > 0$$

  we see that also the fourth order Charbonnier diffusion always performs forward diffusion. With the observation

  $$\left(\lambda^2 + s^2\right)^{-\frac{1}{2}} = \lambda \left(1 + \frac{s^2}{\lambda^2}\right)^{-\frac{1}{2}}$$

  it is clear that regularised TV flow [79] using $g(s^2) = (\lambda^2 + s^2)^{-\frac{1}{2}}$ behaves in the same way.

- **Boundary Case:**
  TV flow [4] comes from the diffusivity $g(s^2) = \frac{1}{|s|}$. At all points where the argument s is nonzero we have $\Phi_1(s) = \Phi_2(s) = 0$ which legitimates to consider TV flow as the boundary case between forward and backward diffusion.

- **Forward and Backward Diffusion:**
  The diffusivity function $g(s^2) = (1 + s^2/\lambda^2)^{-1}$ proposed by Perona and Malik [154] leads to the conditions

  $$\begin{aligned} \Phi_1(s) &= \frac{1}{\lambda^4}\left(1+\frac{s^2}{\lambda^2}\right)^{-3}\left(s^2 - 3\lambda^2\right) < 0 \quad \Longleftrightarrow \quad |s| < \sqrt{3}\lambda \\ \Phi_2(s) &= \left(1+\frac{s^2}{\lambda^2}\right)^{-2}\left(1-\frac{s^2}{\lambda^2}\right) > 0 \quad \Longleftrightarrow \quad |s| < \lambda \ . \end{aligned}$$

This really displays the adaptive nature of this diffusivity: Depending on the parameter $\lambda > 0$, the absolute value of the curvature $|\partial_x^2 u|$ leads to forward or backward diffusion. New to the fourth order case is the presence of two conditions and the possibility that only one of them holds, namely in regions where $\lambda < |\partial_x^2 u| < \sqrt{3}\lambda$. Similar conditions hold for the diffusivity $g(s^2) = \exp\left(-s^2/(2\lambda^2)\right)$ also proposed by Perona and Malik [154].

- **Backward Diffusion:**
  The balanced forward-backward diffusivity [109] defined by $g(s^2) = 1/s^2$ leads to $\Phi_1(s^2) = s^{-4} > 0$ and $\Phi_2(s^2) = -s^{-2} < 0$ which implies that it always performs backward diffusion. As for total variation diffusivity we also suppose that the argument is nonzero here.

We conclude that even in the fourth order case there are diffusivities covering the whole spectrum from pure forward to pure backward diffusion. Of special interest are the two diffusivities by Perona and Malik since they allow for adaptive forward and backward diffusion depending on the local absolute value of the second derivative.

Generalisation to Higher Derivative Orders

After the generalisation of this considerations from order two to four, the natural question is if this also works for orders higher than four. Performing the same calculations as above for the sixth order equation

$$\partial_t u \;=\; \partial_x^3 \left(g\left((\partial_x^3 u)^2 \right) \partial_x^3 u \right) \tag{2.42}$$

one obtains the equivalent equation

$$\begin{aligned} \partial_t u \;=\;& \left(2g^{(1)} (\partial_x^3 u)^2 + g^{(0)} \right) \partial_x^6 u \\ &+2 \left(4g^{(3)} (\partial_x^3 u)^4 (\partial_x^4 u)^2 + 12 g^{(2)} (\partial_x^3 u)^2 (\partial_x^4 u)^2 + 3 g^{(1)} (\partial_x^4 u)^2 \right. \\ &\qquad \left. +6g^{(2)} (\partial_x^3 u)^3 (\partial_x^5 u) + 9 g^{(1)} (\partial_x^3 u)(\partial_x^5 u) \right) \partial_x^4 u \;. \end{aligned} \tag{2.43}$$

Since all derivatives of the diffusivity depend on the same argument $(\partial_x^3 u)^2$ this has been omitted here, and we write $g^{(j)} := g^{(j)}((\partial_x^3 u)^2)$ for better readability. The problem now arises from the terms in the last line, since the derivatives $\partial_x^3 u$ and $\partial_x^5 u$ appear here with odd exponents. Thus these summands might have arbitrary sign. This makes it impossible to distinguish several cases as one could do for second and fourth order equations. Even though we have explicitly shown here only the case of the sixth order equation for illustration, we have checked the equations up to a derivative order of twelve: They all comprehend terms where the sign can not be determined.

This does not meen that there the numerical behaviour of higher order filters would be not plausible. We have already mentioned above that the nonlinear

factor in front of the highest derivative order is always the same as in the second order case. This means for all $p \in \mathbb{N}$, it is possible to write

$$\begin{aligned} \partial_t u &= (-1)^{p+1} \partial_x^p \left(g\left((\partial_p^p u)^2 \right) \partial_x^p u \right) && (2.44) \\ &= (-1)^{m+1} \Phi'\left((\partial_x^{2p} u)^2 \right) \partial_x^{2p} u + R(u, \Psi, p) && (2.45) \end{aligned}$$

with $\Phi(s^2) = 2g'(s^2)s^2 + g(s^2)$. The term $R(u, \Psi, p)$ involves only derivatives of u with orders smaller than the maximal order $2p$. Even if it is not possible to transfer the complete reasoning to the higher order equation, one can see that at least the factor in front of the largest diffusivity order term behaves in the same way than the one for the second order.

2.2.4 Discrete Nonlinear Filtering

Let us now turn our attention to discretisation approaches for the continuous models we have seen so far. To this end, we consider the vectors $f, u \in \mathbb{R}^N$ with $N \in \mathbb{N}$ pixels. There are $R \in \mathbb{N}$ matrices $D_r \in \mathbb{R}^{M \times N}$, $r = 1, \ldots, R$ which extract the relevant features for penalisation of our discrete image u. For discretising the filters described so far, the matrices D_r approximate all partial derivatives of u. We consider energy functions of the form

$$E(u) := \sum_{i=1}^{N} (u_i - f_i)^2 + \alpha \sum_{i=1}^{M} \Psi\Big(\sum_{r=1}^{R} ((D_r u)_i)^2 \Big) \ . \tag{2.46}$$

Note that at this stage we did not specify the arrangement of pixels, and also the matrices D_r are just assumed to have real entries. We notice that by doing so, the energy function (2.46) can handle an image domain with several dimensions as well as multi-channel images. After showing the general procedure, we will give some examples where all the details are specified explicitly.

To determine a minimum of E, we are interested in critical points u with $\nabla E(u) = 0$. Calculating the gradient yields the necessary condition for a minimiser

$$\frac{u - f}{\alpha} = -\sum_{r=1}^{R} D_r^T \, \Phi_D(u) \, D_r u \tag{2.47}$$

where $\Phi_D(u)$ is defined as diagonal matrix

$$\Phi_D(u) := \operatorname{diag}\left(\Psi'\Big(\sum_{r=1}^{R} ((D_r u)_j)^2 \Big) \right)_{j=1,\ldots,M} \ . \tag{2.48}$$

We are going to write $g(s) = \Psi'(s)$ in the following, since Ψ' plays the role of a diffusivity. Analogue as in the continuous case, we can again see the left-hand side as forward difference approximating a first derivative in an artificial time variable

with a time step size $t = \alpha$. Equation (2.47) is then an implicit discretisation for the nonlinear ordinary differential equation

$$\partial_t u \;=\; -\sum_{r=1}^{R} D_r^T \, \Phi_D(u) \, D_r u \;\; . \tag{2.49}$$

Let us choose a time step size $\tau > 0$. We describe three common ways to discretise (2.49) in time:

Explicit Euler Forward Discretisation: The simplest scheme is given as

$$\begin{aligned} u^0 &= f \\ u^{k+1} &= u^k - \tau \sum_{r=1}^{R} D_r^T \, \Phi_D(u^k) \, D_r u^k \end{aligned} \tag{2.50}$$

where $k \in \mathbb{N}$ denotes the iteration index. It is called explicit since we can calculate the variable at the new time step as matrix-vector multiplication.

Semi-Implicit Discretisation: In the semi-implicit scheme, we calculate the matrix with the help of the known vector u^k:

$$u^{k+1} \;=\; \left(I + \tau \sum_{r=1}^{R} D_r^T \, \Phi_D(u^k) \, D_r \right)^{-1} u^k \;\; . \tag{2.51}$$

One has to solve a linear system of equations in order to obtain the next iterand.

Implicit Discretisation: For completeness, we also mention the fully implicit discretisation

$$u^k \;=\; \left(I + \tau \sum_{r=1}^{R} D_r^T \, \Phi_D(u^{k+1}) \, D_r \right)^{-1} u^{k+1} \tag{2.52}$$

where also the matrix is determined from the new values u^{k+1}. This scheme requires in each iteration the solution of a nonlinear system of equations and is thus computationally most expensive.

Properties of the Discrete Filters

In the following, we are going to investigate the basic properties of the discrete filtering methods. For a matrix $A \in \mathbb{R}^{M \times N}$, let $\|A\|$ denote the spectral norm

$$\|A\| \;=\; \max \left\{ \sqrt{|\lambda|} \mid \lambda \text{ eigenvalue of } A^T A \right\} \;\; .$$

We are going to use the fact that the spectral norm is the corresponding matrix norm to the ℓ^2 vector norm (see [100, 177], for example). We will see that in order to achieve ℓ^2-stability with an explicit scheme, we have to choose the time step size τ relatively small while the semi-implicit scheme and the implicit scheme allow arbitrary large time step sizes:

Proposition 2.24 (ℓ^2-Stability Condition)
The explicit discretisation scheme (2.50) is stable in the ℓ^2-norm if the time step size τ satisfies the condition

$$\tau \;\leq\; 2 \left(\sup_{s \in \mathbb{R}} g(s) \sum_{r=1}^{R} \|D_r\|^2 \right)^{-1} \quad . \tag{2.53}$$

The semi-implicit discretisation (2.51) and the implicit discretisation (2.52) are stable in the ℓ^2-norm for arbitrary time step sizes $\tau > 0$.

Proof: From definition (2.48) and the property $g(s) = \Psi'(s) \geq 0$ for all $s \in \mathbb{R}$ we see that $\Phi_D(u)$ is positive semi-definite for arbitrary $u \in \mathbb{R}^N$. The symmetric multiplication with D_r does not change this property, and so $D_r^T \Phi_D(u^k) D_r$ is positive semi-definite and symmetric for all r. Since the set of all positive semi-definite matrices is closed under addition, the whole sum $\sum_{r=1}^{R} D_r^T \, \Phi_D(u^k) \, D_r$ has eigenvalues in the interval

$$\left[0, \; \sup_{s \in \mathbb{R}} g(s) \sum_{r=1}^{R} \|D_r\|^2 \right] \quad .$$

The complete matrix of the explicit scheme

$$I - \tau \sum_{r=1}^{R} D_r^T \, \Phi_D(u^k) \, D_r$$

then has eigenvalues in the interval

$$\left[1 - \tau \sup_{s \in \mathbb{R}} g(s) \sum_{r=1}^{R} \|D_r\|^2, \; 1 \right] \quad .$$

If the condition

$$1 - \tau \sup_{s \in \mathbb{R}} g(s) \sum_{r=1}^{R} \|D_r\|^2 \;\geq\; -1 \quad \Longleftrightarrow \quad \tau \;\leq\; 2 \left(\sup_{s \in \mathbb{R}} g(s) \sum_{r=1}^{R} \|D_r\|^2 \right)^{-1} \tag{2.54}$$

is satisfied, the scheme is ℓ^2-stable.

There is no such restriction for the corresponding semi-implicit and implicit schemes: The matrix

$$I + \tau \sum_{r=1}^{R} D_r^T \, \Phi_D(u) \, D_r$$

has eigenvalues larger or equal to one for all vectors $u \in \mathbb{R}^N$. The eigenvalues of its inverse thus lie in the interval $(0, 1]$, and the ℓ^2-norm of the image u^k can not increase. □

Besides these different restrictions due to numerical stability, the following property is common to all three discretisation methods: We now give the discrete

analogon to Proposition 2.23 concerning moment preservation. In the discrete case, the statement is formulated slightly more general, since we have used general matrices in our discrete regularisation and diffusion approaches.

Proposition 2.25 (Invariant Subspace)
The discrete necessary condition (2.47) for a minimiser, the corresponding semi-discrete equation (2.49) as well as the fully discrete schemes (2.50), (2.51) and (2.52) leave the subspace $\bigcap_{r=1}^{R} \ker(D_r)$ *of* $\mathbb{R}^N$ *invariant.*

Proof: It is well-known from linear algebra that $\operatorname{ran} A^T = (\ker A)^\perp$ for all real matrices $A \in \mathbb{R}^{M\times N}$. For our matrices D_r, it follows that

$$\operatorname{ran} \sum_{r=1}^{R} D_r^T \Phi_D(u) D_r \subseteq \bigcup_{r=1}^{R} \operatorname{ran} D_r^T = \bigcup_{r=1}^{R} (\ker D_r)^\perp \subseteq \left(\bigcap_{r=1}^{R} \ker D_r \right)^\perp .$$

For the above mentioned schemes and equations this means that the changes can only affect the subspace orthogonal to the intersection of all kernels. □

When our matrices implement appropriate finite difference approximations of derivatives, the point evaluations of polynomials are in the kernels of all D_r. In this case, this proposition ensures the preservation of discrete moments up to a certain order. In that sense, it is a discrete analogue to the Propositions 2.17 and 2.23. The boundary conditions in the discrete setting also have a strong influence with respect to the question whether the polynomials are in the kernel or not, see Figure 2.10 for an example and [56] for details in the one-dimensional case.

In the following, we give some examples how the matrices D_r can be chosen to implement filtering methods in practice, for example with (2.50) or (2.51). We have restricted ourselves here to finite difference derivative approximations since they are acting locally. An alternative would have been to use spectral methods ([186, 56], for example) that obtain a global estimate for the derivative. Nevertheless, they are often not so suitable for non-smooth signals. Since our goal is edge preservation and enhancement, we use finite differences here.

Example 2.26 (General Filter of Order 2p in One Dimension)
We consider discretisations for the filter class introduced in Example 2.18:

Natural boundary conditions: For natural boundary conditions, the strategy is only to approximate the derivatives in the smoothness term at those points of the pixel grid where the whole approximation stencil still fits in. This does not impose any conditions at the boundary, and in this sense the necessary conditions as described above then lead to natural boundary conditions. Let $h > 0$ denote the spatial step size. In the one-dimensional case, the multiplication of

the matrix

$$D_{1,N}^{\mathcal{N}} := \frac{1}{h} \begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(N-1)\times N} \tag{2.55}$$

with a vector yields an approximation of the first derivative. The superscript $\mathcal{N}$ stands for natural boundary conditions here while the subscripts denote the derivative order and the number of pixels. For higher derivatives, we can simply use the corresponding products

$$D_{p,N}^{\mathcal{N}} = D_{1,N-p+1}^{\mathcal{N}} \cdot \ldots \cdot D_{1,N}^{\mathcal{N}} . \tag{2.56}$$

We note that $D_{p,N}^{\mathcal{N}} \in \mathbb{R}^{(N-p)\times N}$ is not a quadratic matrix. The kernel of $D_{p,N}^{\mathcal{N}}$ is given by point evaluations of polynomials of degree $p-1$. An explicit scheme for one-dimensional nonlinear diffusion of order $2p$ reads as

$$u^{k+1} = u^k - \tau \left(D_{p,N}^{\mathcal{N}}\right)^T \Phi_{D_{p,N}^{\mathcal{N}}}(u^k) D_{p,N}^{\mathcal{N}} u^k \quad k \in \mathbb{N} \tag{2.57}$$

with $u^0 = f$. It realises a discretisation with natural boundary conditions; details can be found in [56]. Since $\|D_{p,N}^{\mathcal{N}}\| \leq 2^p/h^p$, we see with Proposition 2.24 that the scheme is ℓ^2-stable for time step sizes

$$\tau \leq \frac{h^{2p}}{\sup_{s\in\mathbb{R}} g(s)\, 2^{2p-1}} .$$

As a typical example, let us assume a spatial step size $h = 1$ and choose the Perona-Malik diffusivity g with $|g| \leq 1$. For an explicit discretisation, one has to choose $\tau \leq 1/2$ for order $p = 1$, $\tau \leq 1/8$ for $p = 2$, and $\tau \leq 1/32$ for $p = 3$ in this case.

Periodic boundary conditions: Since it will be useful in the next chapter, we also write down the discretisation for diffusion with periodic boundary conditions. We use the circulant matrix

$$D_{1,N}^{\mathcal{P}} := \frac{1}{h} \begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & -1 & 1 \\ 1 & 0 & \dots & 0 & -1 \end{pmatrix} \in \mathbb{R}^{N\times N} \tag{2.58}$$

to approximate the first derivative and its p-th power

$$D_{p,N}^{\mathcal{P}} := \left(D_{1,N}^{\mathcal{P}}\right)^p \tag{2.59}$$

for higher derivative orders. Here, $\mathcal{P}$ denotes periodic boundary conditions. We notice that for larger values of p the higher derivative approximations are obtained at shifted positions by half of the derivative order. This does not influence the correctness of the result since for the outer derivative in the diffusion equation, we use the transposed matrix which shifts in the other direction by exactly the same amount.

The limits for the time step size are the same as for natural boundary conditions. We would like to mention that independent of the order p, the matrix $D_{p,N}^{\mathcal{P}}$ always has the kernel $\operatorname{span}\{(1,\dots 1)^T\}$ of constant signals.

The effect of these different boundary conditions and the resulting different kernels of the matrices has already become visible with the filtering example in Figure 2.10.

Example 2.27 (Fourth-Order Filtering in 2-D)
An approach how to discretise the fourth order PDE in 2-D shown in Example 2.20 has been given by Lysaker et al. [122, 123]. They have used small one-sided stencils with only four pixels for the approximation of the mixed derivatives. We use another way here by using symmetric stencils since it has shown visually good results in practice. In 2-D, we do not write down the discretisations in matrix form, but in stencil notation. For the second-order derivatives with respect to one variable we use the standard second differences with spatial step sizes $h_x, h_y > 0$ in x- and y-direction:

$$u_{xx} \;\approx\; \frac{1}{h_x^2}\begin{array}{|c|c|c|}\hline 1 & -2 & 1 \\ \hline\end{array} \cdot u \;, \qquad u_{yy} \;\approx\; \frac{1}{h_y^2}\begin{array}{|c|}\hline 1 \\ \hline -2 \\ \hline 1 \\ \hline\end{array} \cdot u \;. \tag{2.60}$$

It is well-known that the corresponding matrices D_{xx} and D_{yy} satisfy the property $\|D_{xx}\| \leq \frac{2}{h_x^2}$ and $\|D_{yy}\| \leq \frac{2}{h_y^2}$, respectively. For the mixed differences, there are several possibilities of discretisation. For the mixed derivatives, we use the following stencils:

$$u_{xy} \;\approx\; \frac{1}{2h_xh_y}\begin{array}{|c|c|c|}\hline 0 & -1 & 1 \\ \hline -1 & 2 & -1 \\ \hline 1 & -1 & 0 \\ \hline\end{array} \cdot u \;, \qquad u_{yx} \;\approx\; \frac{1}{2h_xh_y}\begin{array}{|c|c|c|}\hline -1 & 1 & 0 \\ \hline 1 & -2 & 1 \\ \hline 0 & 1 & -1 \\ \hline\end{array} \cdot u \;. \tag{2.61}$$

At the example of the first stencil for u_{xy} we show how to obtain limits for the norm of the matrix with Gershgorin's theorem (see [100], for example): Let us denote the matrix corresponding to the approximation of u_{xy} with D_{xy}. Applying the stencil twice gives

$$\frac{1}{4h_x^2h_y^2}\begin{array}{|c|c|c|c|c|}\hline & & 1 & -2 & 1 \\ \hline & 2 & -6 & 6 & -2 \\ \hline 1 & -6 & 10 & -6 & 1 \\ \hline -2 & 6 & -6 & 2 & \\ \hline 1 & -2 & 1 & & \\ \hline\end{array} \cdot u \;. \tag{2.62}$$

j+2	$\frac{1}{4h_x^2h_y^2}g_{i-1j+1}$	$\frac{-1}{4h_x^2h_y^2}(g_{i-1j+1}+g_{ij+1})$	$\left(\frac{1}{h_y^4}+\frac{1}{2h_x^2h_y^2}\right)g_{ij+1}$	$\frac{-1}{4h_x^2h_y^2}(g_{ij+1}+g_{i+1j+1})$	$\frac{1}{4h_x^2h_y^2}g_{i+1j+1}$
j+1	$\frac{-1}{4h_x^2h_y^2}(g_{i-1j+1}+g_{i-1j})$	$\frac{1}{2h_x^2h_y^2}(g_{ij}+g_{ij+1}$ $+g_{i-1j}+g_{i-1j+1})$	$\frac{-2}{h_y^4}(g_{ij+1}+g_{ij})$ $+\frac{-1}{h_x^2h_y^2}(g_{ij}+g_{ij+1})$ $\frac{-1}{4h_x^2h_y^2}(g_{i-1j+1}+g_{i+1j+1}$ $+g_{i-1j}+g_{i+1j})$	$\frac{1}{2h_x^2h_y^2}(g_{ij}+g_{ij+1}$ $+g_{i+1j}+g_{i+1j+1})$	$\frac{1}{4h_x^2h_y^2}(g_{i+1j+1}+g_{i+1j})$
j	$\left(\frac{1}{h_x^4}+\frac{1}{2h_x^2h_y^2}\right)g_{i-1j}$	$\frac{-2}{h_x^4}(g_{i-1j}+g_{ij})$ $-\frac{1}{h_x^2h_y^2}(g_{ij}+g_{i-1j})$ $-\frac{1}{h_x^2h_y^2}(g_{i-1j-1}+g_{i-1j+1}$ $+g_{ij-1}+g_{ij+1})$	$\frac{1}{h_x^4}(g_{i-1j}+4g_{ij}+g_{i+1j})$ $+\frac{1}{h_y^4}(g_{ij-1}+4g_{ij}+g_{ij+1}$ $+\frac{1}{4h_x^2h_y^2}(g_{i-1j-1}+g_{i-1j+1}$ $+g_{i+1j-1}+g_{i+1j+1})$ $+\frac{1}{2h_x^2h_y^2}(g_{i-1j}+g_{i+1j}$ $g_{ij-1}+g_{ij+1})$ $+2h_x^2h_y^2g_{ij}$	$\frac{-2}{h_x^4}(g_{ij}+g_{i+1j})$ $-\frac{1}{h_x^2h_y^2}(g_{ij}+g_{i+1j})$ $-\frac{1}{h_x^2h_y^2}(g_{i+1j-1}+g_{i+1j+1}$ $+g_{ij-1}+g_{ij+1})$	$\left(\frac{1}{h_x^4}+\frac{1}{2h_x^2h_y^2}\right)g_{i+1j}$
j-1	$\frac{-1}{4h_x^2h_y^2}(g_{i-1j}+g_{i-1j-1})$	$\frac{1}{2h_x^2h_y^2}(g_{ij}+g_{ij-1}$ $+g_{i-1j}+g_{i-1j-1})$	$\frac{-2}{h_y^2}(g_{ij}+g_{ij-1})$ $\frac{-1}{h_x^2h_y^2}(g_{ij}+g_{ij-1})$ $\frac{1}{4h_x^2h_y^2}(g_{i-1j-1}+g_{i+1j-1}$ $+g_{i-1j}+g_{i+1j})$	$\frac{1}{2h_x^2h_y^2}(g_{ij}+g_{ij-1}$ $+g_{i+1j}+g_{i+1j-1})$	$\frac{-1}{4h_x^2h_y^2}(g_{i+1j}+g_{i+1j-1})$
j-2	$\frac{1}{4h_x^2h_y^2}g_{i-1j-1}$	$\frac{-1}{4h_x^2h_y^2}(g_{i-1j-1}+g_{ij-1})$	$\left(\frac{1}{h_y^2}+\frac{1}{2h_x^2h_y^2}\right)g_{ij-1}$	$\frac{-1}{4h_x^2h_y^2}(g_{i+1j-1}+g_{ij-1})$	$\frac{1}{4h_x^2h_y^2}g_{i+1j-1}$
	$i-2$	$i-1$	i	$i+1$	$i+2$

Table 2.1: Stencil for fourth order nonlinear diffusion filtering.

In this stencil notation, we see the entries of one row of the corresponding matrix $D_{xy}^T D_{xy}$. The sum of all absolute values of these entries without the diagonal entry is 54. Gershgorin's theorem shows that the corresponding eigenvalues are in the interval $\left[-\frac{54}{4h_x^2h_y^2}, \frac{64}{4h_x^2h_y^2}\right]$. This means the spectral norm of D_{xy} is less than or equal to $\frac{4}{h_xh_y}$.

Together, for such a discretisation, we obtain a limit for the time step size

$$\tau \;\leq\; \left(\sup_{s\in\mathbb{R}} g(s)\left(\frac{2}{h_x^2}+\frac{2}{h_y^2}+2\cdot\frac{16}{h_xh_y}\right)\right)^{-1} \tag{2.63}$$

in order that the explicit scheme is stable. For the case $h_x = h_y = 1$, we obtain

$$\tau \;\leq\; \left(36\cdot\sup_{s\in\mathbb{R}} g(s)\right)^{-1} \quad . \tag{2.64}$$

This means that we have to choose the time step size $\tau \leq \frac{1}{36}$ if we want to use the Perona-Malik diffusivity. For regularised total variation with typical values of $\lambda = 0.01$, one has the even smaller limit $\tau \leq \frac{\lambda}{3600}$.

For practical purposes, this limitation for the maximal time step size is severe, and we show how to derive semi-implicit discretisations to overcome this drawback. In Table 2.1 we display the corresponding stencil for fourth order nonlinear diffusion filtering. In the stencil, g_{ij} approximates $g(\|H(u)\|_F^2)$ where the partial derivatives in the Hessian are approximated as described above. To implement natural boundary conditions, we have to set $g_{ij} = 0$ for all (i, j) at the boundary of the discrete grid Ω_h, since here we do not have enough data to approximate $\|H(u)\|_F$. This principle has already been described in the 1-D setting. If A denotes the matrix corresponding to the stencil in Table 2.1, we have to solve the linear system of equations

$$(I + \tau A)u^{k+1} = u^k \quad . \tag{2.65}$$

As solver we use successive over-relaxation (SOR). We do not give a full description of this method here, since it can be found in many textbooks on numerical methods, for example [177, 167, 160]. In the proof of Proposition 2.24, we have seen that $I + \tau A$ is positive definite. This guarantees the convergence of the SOR method with the theorem of Ostrowski and Reich (see [177, p. 631], for example). In practice, time step sizes τ in the order from 1 to 5 give visually good results. SOR introduces two further numerical parameters: the number of iterations and the relaxation factor ω. Usually we have worked with a relaxation factor $\omega = 1.5$ and with 25 to 50 iterations. In practice this choice of parameters was sufficient for small residues and visually good results.

Total Variation Regularisation and Splines

With the examples given above we have a formal description for discrete regularisation and diffusion filters. The properties of leaving higher moments unchanged already indicate that the filtering results are closely connected to polynomials. In the special case of discrete regularisation with the ℓ^1-norm as penalising function, one can even formalise these connections [174, 175, 173]: It can be shown that the results are discrete splines as described by Mangasarian and Schumaker [130, 131]. In that sense, ℓ^1-regularisation yields discrete spline approximations of the given data where the number and position of spline knots are determined adaptively by the given data and the regularisation weight.

One can see that the knots can be found as contact points of a taut string with fixed end points within a tube of width α around the discrete p-th integrand of the initial data. For the minimisation procedure, one can use the very efficient taut-string algorithm as introduced by Mammen and van de Geer [129]. Further information about these so-called tube methods can be found in [54, 98].

There is no proof for similar properties in the case of the Perona-Malik diffusivity: This case is more complicated to describe since the corresponding penaliser is not convex and has an additional contrast parameter. However, the numerical experiments in the next section indicate that with a suitable choice of the scale parameter, one can also obtain results that can be segmented in – at least approximately – polynomial regions.

2.2.5 Numerical Experiments

In this subsection we show results of higher order nonlinear diffusion filtering in one and two dimensions with different orders.

In our first experiment, we consider a one-dimensional Gaussian signal and filtering results for the orders two, four and six as displayed in Figure 2.11. For $2p$-th order filtering with $g(s^2) = (1 + s^2/\lambda^2)^{-1}$ the parameter λ is chosen such that there are regions with $|\partial_x^p u| > \sqrt{3}\lambda$. We have seen in Subsection 2.2.3 that this is expected to yield backward diffusion, at least for the orders two and four. While second order filtering yields enhancement of edges, the fourth order filtering result tends to be piecewise linear with enhanced curvature at corner points. This observation for fourth order filtering is further affirmed by the almost piecewise constant derivative approximation of the filtering result also shown in Figure 2.11. The sixth order filter behaves analogously yielding a piecewise quadratic signal which can also be seen by its piecewise constant approximation of the second derivative.

Together, this experiment corroborates the belief that Perona-Malik diffusion with order $2p$ yields for an appropriate choice of the parameter λ a filtered signal which is polynomial of degree $p - 1$ on disjoint intervals. Experimentally high values of the first or second order derivative of the initial signal determine

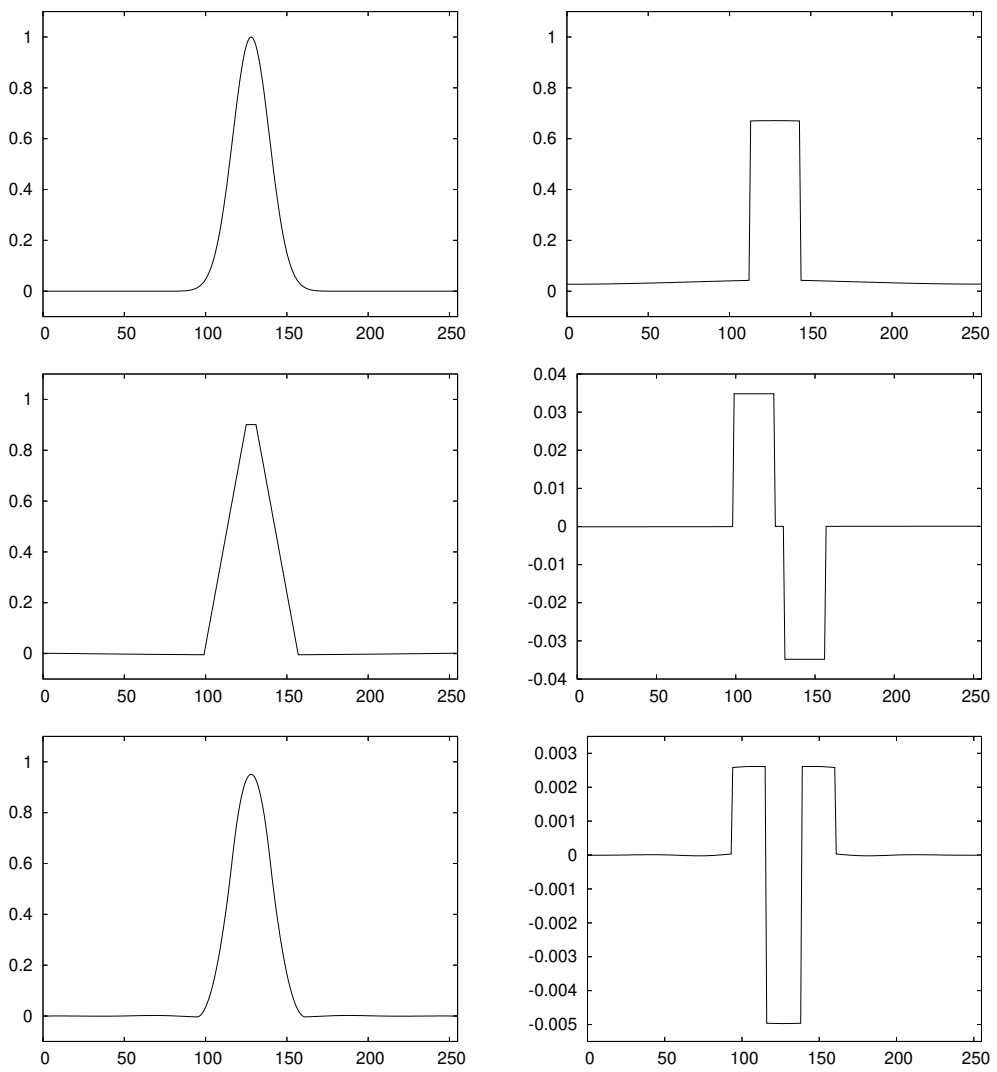


Figure 2.11: *Top left:* Gaussian signal. *Top right:* Second order Perona-Malik filtering. *Second row: Left:* Fourth order Perona-Malik filtering. *Right:* First derivative of fourth order filtering result. *Bottom left:* Sixth order Perona-Malik filtering. *Bottom right:* Second derivative of sixth order filtering result.

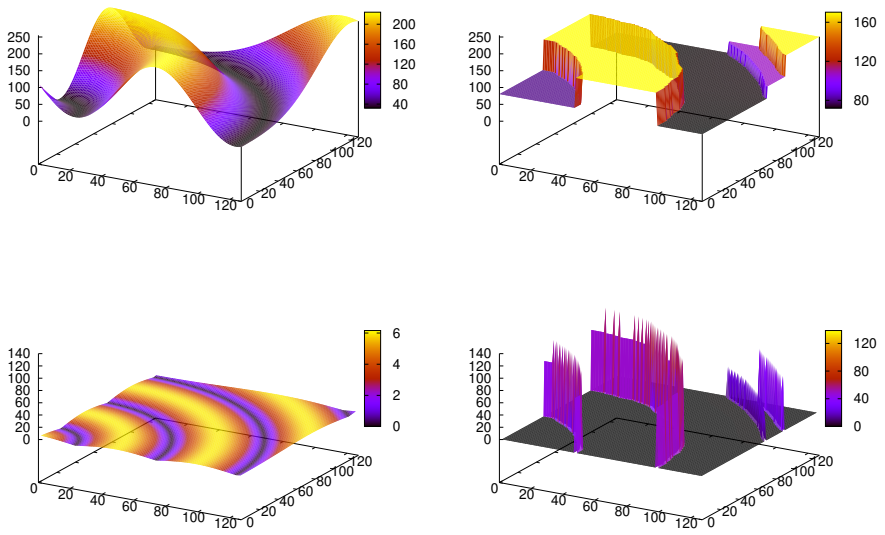


Figure 2.12: Second order Perona-Malik filtering and edge enhancement. *Top left:* Original image, 128×128 pixels. *Top right:* Second order Perona-Malik filtering. *Bottom:* Corresponding gradient norms.

the location of the interval boundaries where the filtered signal is $p - 2$ times continuously differentiable.

Our second experiment shows that a similar adaptive behaviour is also possible in two dimensions. The Figures 2.12 and 2.13 show some plots where an image is seen as surface in the three-dimensional space, and the grey values determine the z component. In Figure 2.12 we see the original image and the result for edge-enhancing Perona-Malik filtering. To better visualise the edge-enhancement, the norm of the gradient is displayed for both the original and filtered image.

Similar results can be seen for fourth order filtering, too: Figure 2.13 displays the same original image and a filtered version with fourth order Perona-Malik filtering. Here the relevant feature is not the gradient norm, but the Frobenius norm of the Hessian. We see that this norm becomes zero almost everywhere. There are only some lines in the image where the Hessian norm even is increased. At these lines the curvature is enhanced strongly.

Let us now show the results of some denoising experiments. For the first experiment, we use the test image already used in Figure 1.1. For better visibility, we only display a section of size 128×128 pixels, but all calculations have been performed with the whole images. We compare the results for second and fourth order nonlinear diffusion to see if higher order filters are in fact helpful to avoid staircasing. We have used semi-implicit schemes for all filters. Figure 2.14

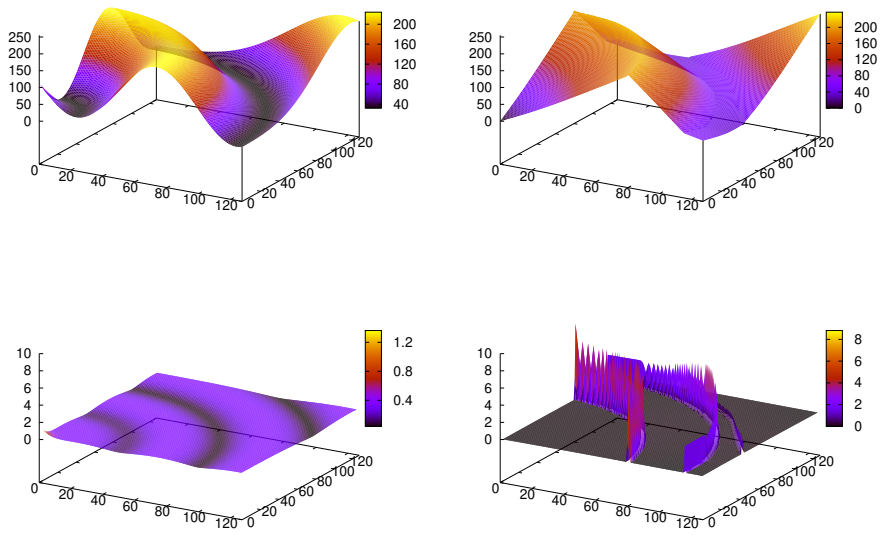


Figure 2.13: Fourth order Perona-Malik filtering and piecewise linearity. *Top left:* Original image, 128×128 pixels. *Top right:* Fourth order Perona-Malik filtering. *Bottom:* Corresponding Frobenius norms of the Hessian.

shows the corresponding results for the regularised total variation diffusivity. As regularisation parameter we have used $\lambda = 0.01$, and experiments with smaller values have shown that the results show no visible or measurable changes. The results have been optimised in order to minimise the ℓ^1- and ℓ^2-error. The resulting error norms and parameters can be found in Table 2.2. The second-order filtering results show staircasing artifacts. In the fourth-order case, minimising the ℓ^1-error leads to unsharp edges, while the smooth grey value transitions are preserved well. With the gradient norm approximations, the staircasing for the second order filtering is clearly visible, while the fourth order filter reconstructs the linear grey value transition better. Using the Perona-Malik diffusivity, we obtain the results shown in Figure 2.15. In general, we see that the edges are much better preserved with this diffusivity. This is also reflected by smaller error measures. It is interesting to see that for the Perona-Malik filter of second order, comparably large values for λ are preferred: This indicates that the staircasing is so strong for smaller values that it increases the error significantly. With the larger values of λ we hardly see any staircasing here even for the second order filter. Nevertheless, the linear grey value transitions are still better recovered with the fourth order method.

The second denoising experiment deals with the real-world test image shown in Figure 2.16. The optimal results in terms of the ℓ^1- and ℓ^2-error for second and

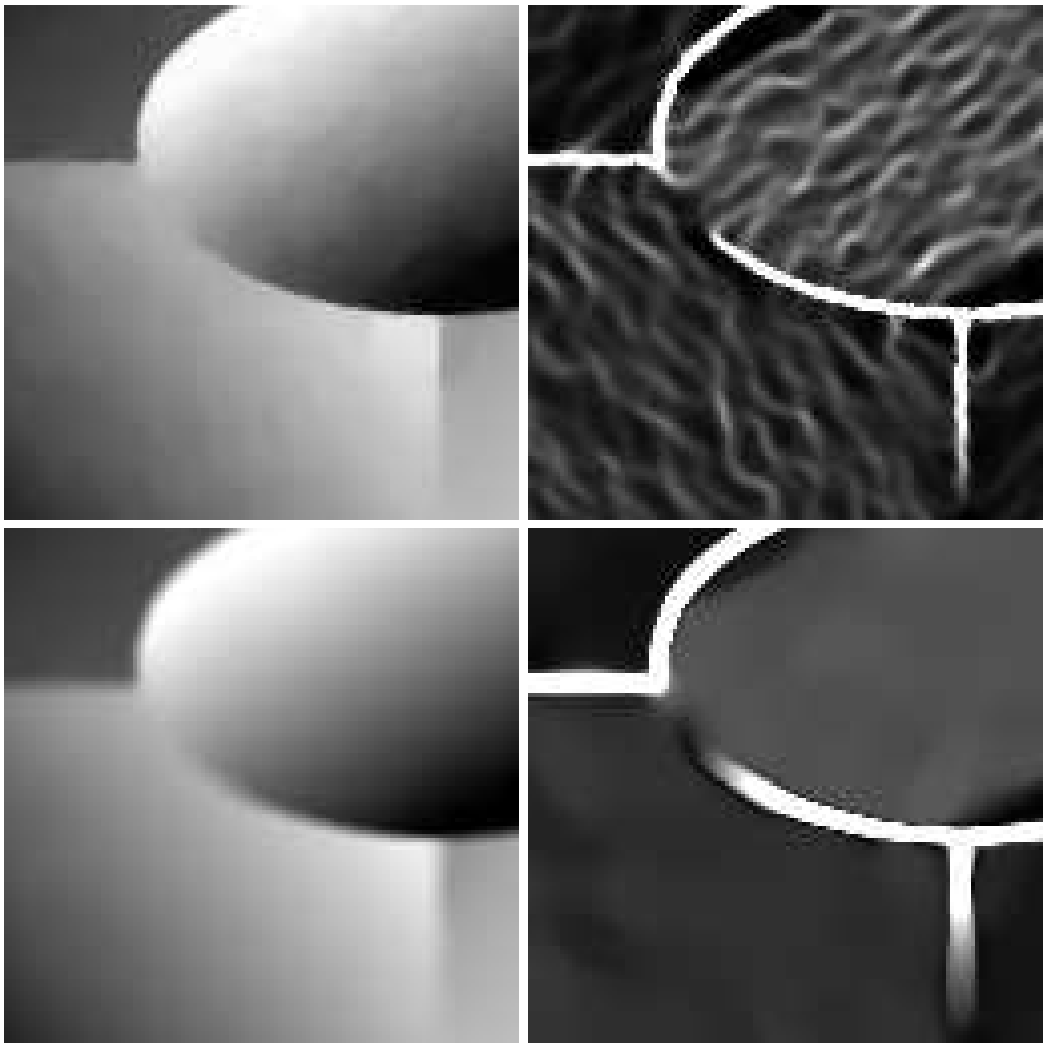


Figure 2.14: Denoising results for regularised total variation diffusivity $g(s^2) = (s^2+\lambda^2)^{-1/2}$ with $\lambda = 0.01$. Parameters optimised for minimal ℓ^1-error. *Top row:* Second order nonlinear diffusion. *Bottom row:* Fourth order nonlinear diffusion. *Left column:* Sections of size 128×128 pixels of the results. *Right column:* Corresponding approximation of the gradient norm.

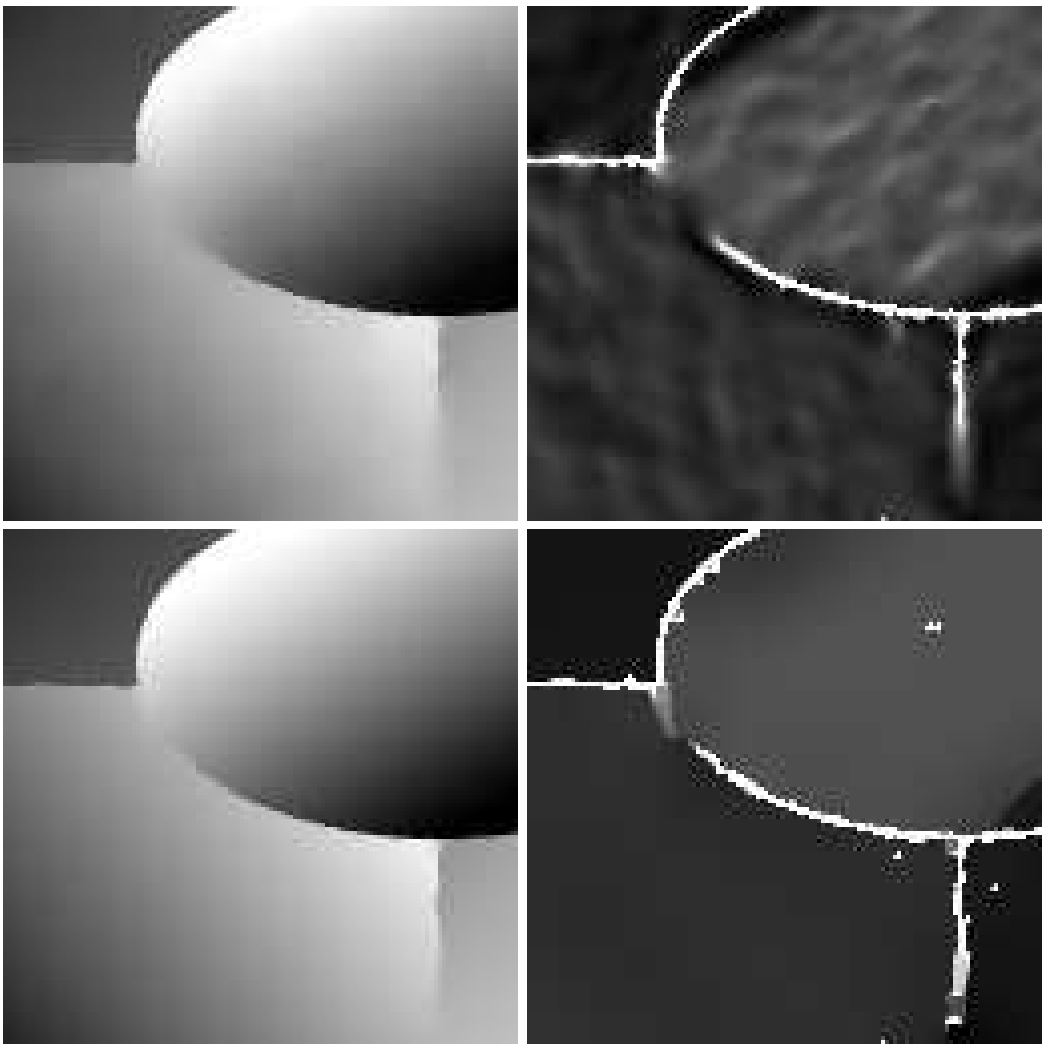


Figure 2.15: Denoising results for Perona-Malik diffusivity $g(s^2) = (1 + s^2/\lambda^2)^{-1}$. Parameters optimised for minimal ℓ^1-error. *Top row:* Second order nonlinear diffusion. *Bottom row:* Fourth order nonlinear diffusion. *Left column:* Sections of the filtering results. *Right column:* Corresponding approximation of the gradient norm.

method	Error in ℓ^1-norm			Error in ℓ^2-norm		
	error	λ	t	error	λ	t
original image	1047936	–	–	5132.04	–	–
reg. TV 2	139950	0.01	15.75	937.94	0.01	20.75
reg. TV 4	144715	0.01	43.00	1329.45	0.01	16.00
PM 2	115503	7.76	12.50	749.31	7.34	12.50
PM 4	**75944.4**	**0.20**	**9274.00**	**682.63**	**0.20**	**10483.00**

Table 2.2: Error measures for artificial denoising example with regularised total variation (reg. TV) and Perona-Malik (PM) diffusivities with orders two and four.

Figure 2.16: Real-world test image. *Left:* Original image, 256×256 pixels. *Right:* With additive Gaussian noise, standard deviation $\sigma = 10$.

fourth order Perona-Malik diffusion can be found in Figure 2.17. Again we only display sections of size 128×128 pixels for better visibility of the differences. For this real-world image, we see that the visible differences are mostly concentrated on the edges: With fourth order diffusion, the edges contain less noise and seem a bit smoother and more natural. This is also reflected by smaller error norms, as it can be seen in Table 2.3.

2.3 Higher Order Data Terms

So far, we have only considered the use of derivatives in the smoothness term of the energy function.

In variational optic flow computation, it is very helpful to use higher derivatives not only in the smoothness term, but also in the data term [151]. In that

Figure 2.17: Denoising of for real-world data with Perona-Malik diffusivity $g(s^2) = (1 + s^2/\lambda^2)^{-1}$. *Top row:* Second order nonlinear diffusion. *Bottom row:* Fourth order nonlinear diffusion. *Left column:* Results with optimal ℓ^1-error. *Right column:* Results with optimal ℓ^2-error.

method	Error in ℓ^1-norm			Error in ℓ^2-norm		
	error	λ	t	error	λ	t
original image	523968	–	–	2566.02	–	–
PM 2	197919	3.90	8.00	1143.16	3.90	8.25
PM 4	**195667**	**1.36**	**47.00**	**1112.91**	**1.79**	**28.50**

Table 2.3: Error measures for real-world denoising example with Perona-Malik (PM) diffusivities and diffusion orders two and four.

context, this models the assumption that not only the grey values of the corresponding pixel itself, but also their first or second derivative stay constant in time during the motion. As experimental results show, this can be very helpful to cope with changes in illumination from frame to frame.

2.3.1 Modelling

The idea to use derivatives in the data term can also be adapted to image processing: For example, one can penalise not only the quadratic difference between the restored and the initial image, but also the difference of their gradient or of higher derivatives. In [59] discrete analoga to the energy functional

$$\mathscr{E}(u) \;=\; \int_\Omega \Big((u-f)^2 + \alpha\,|\nabla u - \nabla f|^2 + \beta \Big(\sum_{|\gamma|=2} w_\gamma |\mathcal{D}^\gamma u|^2 \Big)^{\frac{1}{2}} \Big)\; dx \tag{2.66}$$

are investigated. Similar to TV regularisation [174, 175], one obtains a spline character of the solutions also in this case. The difference is that here we have discrete splines with defect. The numerical examples shown in [59] clearly indicate that this procedure can yield qualitatively better results than standard TV methods. This can be made plausible by the fact that the defects give the splines more freedom to adapt to non-smooth features of the data which can be especially helpful for edge preservation, for example. A modified version of Chambolle's algorithm [33] is introduced in [59] to solve the minimisation problems.

Here, we will present a slightly more general model involving a combination of several higher order data terms. Two possibilities for the numerical implementation will be described in the context of the methods shown in the previous section.

Let us first describe the general idea: We demand similarity in all derivatives up to a certain order $p-1$ and smoothness or order p. In 1-D, a corresponding energy functional looks like:

$$\mathscr{E}(u) \;=\; \int_\Omega \Big(\sum_{k=0}^{p-1} w_k (u^{(k)} - f^{(k)})^2 \;+\; w_p\, \Psi((u^{(p)})^2) \Big)\; dx \;. \tag{2.67}$$

Let us assume that our initial data f and the solution u are sufficiently smooth, and let the integrand of the energy functional be small at one pixel x_0. This means that the function itself and its partial derivatives up to order $p-1$ are similar to the given data at the position x_0. For all points in a neighbourhood of x_0, this can be interpreted as similarity in terms of the Taylor expansion around x_0:

$$u(x) \;=\; \sum_{k=0}^{p-1} \frac{1}{k!} u^{(k)}(x_0)(x-x_0)^k \;+\; \frac{1}{p!} u^{(p)}(\theta)(x-x_0)^p$$

$$\begin{aligned} &\approx \sum_{k=0}^{p-1} \frac{1}{k!} f^{(k)}(x_0)(x-x_0)^k + \frac{1}{p!} u^{(p)}(\theta)(x-x_0)^p + \frac{1}{p!} f^{(p)}(\theta_2)(x-x_0)^p \\ &\approx f(x) \end{aligned}$$

with θ, θ_2 between x_0 and x. The smoothness term penalises high values of the derivative $u^{(p)}$ and thus helps to make one of the error terms in the above calculation smaller. With this reasoning, such an approach uses the derivatives in the data term to favour a higher degree of similarity in a whole neighbourhood. This explains why it can better adapt to characteristic features of the data.

The higher order data terms were mentioned here to sketch a promising direction how to design more general filters within the variational and PDE framework. In the following subsection, we are going to describe two ways how to discretise and implement such methods and show some numerical examples.

2.3.2 Discretisation and Numerical Experiments

A discrete version of the one-dimensional energy functional (2.67) with higher order data terms can be formulated as

$$E(u) \;=\; \sum_{r=0}^{p-1} \sum_{i=1}^{M_k} w_r (D_r u - D_r f)_i^2 + w_p \sum_{i=1}^{M_p} \Psi((D_p u)_i^2) \;. \tag{2.68}$$

Here, the matrices D_r implement derivative approximations of order k as described in the last section. Since these matrices usually are not quadratic, M_r denotes the dimension of the range of D_r. In particular, we set $D_0 := I$, $M_0 = N$, and $w_0 = 1$. A necessary condition for a minimiser of E is given by

$$0 \;=\; \sum_{r=0}^{p-1} D_r^T D_r (u - f) \;+\; w_p D_p^T \Phi_{D_p}(u) D_p u \tag{2.69}$$

where we use the notations for $\Phi_{D_p}(u)$ as in the previous section. Let us now sketch two ways how to solve this equation:

First, one can understand it as the steady state of the partial differential equation

$$\partial_t u \;=\; -\sum_{r=0}^{p-1} D_r^T D_r (u - f) - w_p D_p^T \Phi_{D_p}(u) D_p u \tag{2.70}$$

for t tending to ∞. In the continuous framework, the equivalent to this equations would be a generalised diffusion-reaction equation. A simple explicit discretisation for (2.70) is given by

$$u^{k+1} \;=\; u^k - \tau \sum_{r=0}^{p-1} D_r^T D_r (u^k - f) - \tau w_p D_p^T \Phi_{D_p}(u^k) D_p u^k \;. \tag{2.71}$$

The disadvantage of this discretisation is that we need to calculate the steady state, and thus we need very large stopping times t.

We can rewrite (2.69) as

$$\sum_{r=0}^{p-1} w_r D_r^T D_r (u - f) \;=\; -w_p D_p^T \Phi_{D_p}(u) D_p u \;\; . \tag{2.72}$$

Under the assumption that all our weights w_r are positive, this is equivalent to

$$\frac{u - f}{w_p} \;=\; -\left(\sum_{r=0}^{p-1} w_r D_r^T D_r \right)^{-1} D_p^T \Phi_{D_p}(u^k) D_p u^k \tag{2.73}$$

For this equivalence, note that $D_0 = I$ and that $D_r^T D_r$ is positive semidefinite for all r. Thus the matrix $\sum_{r=0}^{p-1} w_r D_r^T D_r$ is positive definite and invertible. With the approaches discussed in the previous section this equation is related to the iterative scheme

$$u^{k+1} \;=\; u^k - \tau \left(\sum_{r=0}^{p-1} w_r D_r^T D_r \right)^{-1} D_p^T \Phi_{D_p}(u^r) D_p u^k \;\; . \tag{2.74}$$

Since we have set $w_0 = 1$, we know that the eigenvalues of $\sum_{r=0}^{p-1} w_r D_r^T D_r$ are real and larger or equal to 1. This means that the eigenvalues of its inverse are in the interval $[0, 1]$. This means that the scheme is stable with respect to the ℓ^2-norm with the same limits for the time step size τ as calculated in Proposition 2.24 for nonlinear diffusion of order $2p$. With Example 2.26, we have to choose

$$\tau \leq \frac{h^{2p}}{\sum_{s \in \mathbb{R}} g(s) 2^{2p-1}} \;\; .$$

The scheme (2.74) can be seen as an explicit discretisation of higher order nonlinear diffusion where the right-hand side is smoothed by an implicit diffusion step of all lower orders. In other words, the difference between two iteration steps of higher order nonlinear diffusion is smoothed by linear diffusion of lower orders. Besides one way of implementing higher order data terms, this also offers an alternative interpretation for them.

To conclude this section, we display a denoising example with higher order data terms. We have used a two-dimensional analogon to the second discretisation method (2.74) described above. Figure 2.18 displays the resulting images and the corresponding gradient norms. The parameters and error measures are shown in Table 2.4. The case $w_1 = 0$ is the fourth order regularised total variation diffusion described in the previous section. In practical experiments it turned out that Perona-Malik filtering usually could hardly be improved by the higher order data terms: The edges are preserved well with this penaliser without adding

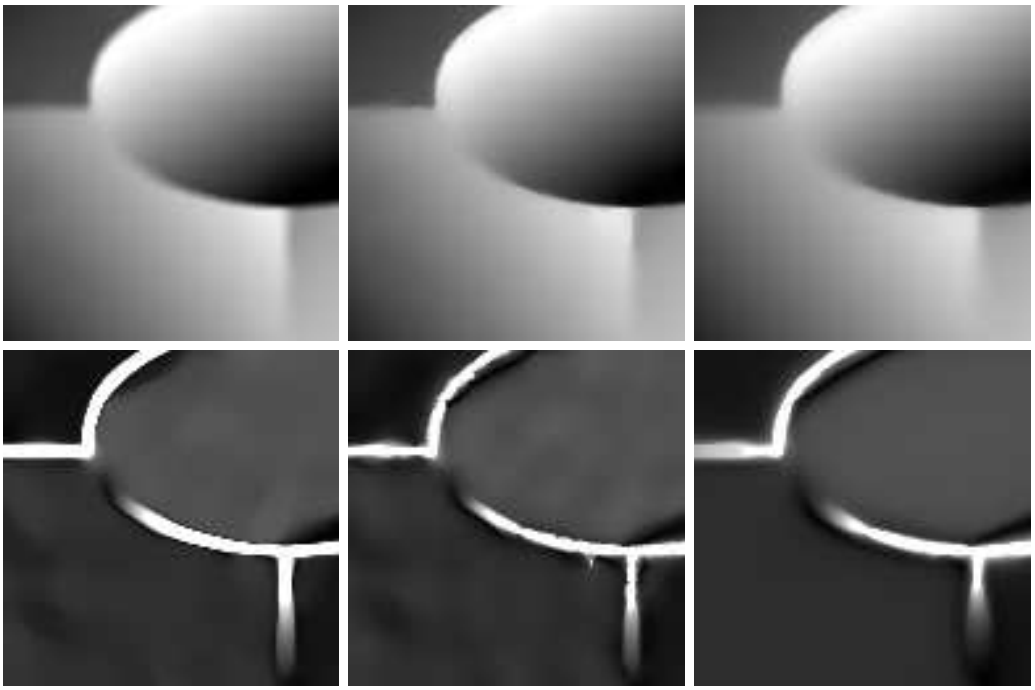


Figure 2.18: Filtering with fourth order regularised total variation and higher order data terms. *Top left:* Without higher order data term. *Top middle:* Weight $w_1 = 5$. *Top right:* $w_1 = 10$. *Bottom row:* Corresponding gradient norm approximations.

further constraints to the data term. Thus we focus on regularised total variation examples since they have shown quite blurred edges in Figure 2.14. We see that the higher order data terms can be helpful to sharpen the edges in this case. This is in accordance with the results presented in [59]. The error measures also reflect the quality enhancement in comparison with regularised total variation filtering.

2.4 Summary

In this chapter, we have discussed variational techniques and corresponding PDE methods for simplification and denoising with a special emphasis on the use of higher and fractional order derivatives.

For linear filtering, we have used linear combinations of fractional powers of the Laplacian in the smoothness term. This choice was made because the corresponding pseudodifferential equations are closely connected to semi-groups and thus satisfy an important property of linear scale-spaces. In fact, we have considered an extension of the classical α-scale-spaces [69] for orders $\alpha > 1$ and for combinations of several orders. Usually the restriction $0 < \alpha \leq 1$ was made in

method	ℓ^1-error	ℓ^2-error	λ	t
original image	523968	2566.02	–	–
$w_1 = 0$	144715	1625.33	0.01	9274.00
$w_1 = 5$	146557	1234.98	0.10	199.86
$w_1 = 10$	**138842**	**1229.22**	**0.01**	**534.65**

Table 2.4: Error measures for artificial denoising example with regularised total variation and higher order data term.

the literature to obtain a maximum-minimum principle. Experimentally we could show that this desirable property can also be achieved by considering a combination of two orders instead of one single order. Necessary for these experiments was a discrete description of the regularisation and diffusion methods. We have further performed experiments to quantify the variance-diminishing properties of our regularisation and diffusion framework.

In the second section, we have turned our attention from linear filtering methods to nonlinear ones to have useful properties such as edge preservation and enhancement. To obtain necessary conditions for the variational problems arising here, we have first applied some methods from the calculus of variations. We have defined a general framework for higher order nonlinear regularisation where all partial derivatives of a given order p are involved in the smoothness term. A motivation for this is the fact that for an ℓ^1-penaliser, one obtains spline approximations as solution [175]. Here, we have focused rather on the corresponding higher order diffusion equations: In terms of forward-backward diffusion, one can see that also higher order filters can not only lead to feature preservation, but also feature enhancement. We have derived stability properties and the preservation of higher moments in the continuous and discrete setting. Numerical experiments have shown that also for the Perona-Malik case solutions can be split in regions where they are approximately of polynomial character. For denoising examples, we have seen that higher order filters can improve the quality of the results. Compared to the regularised total variation diffusivity, using diffusivities with forward-backward behaviour as the Perona-Malik diffusivity has the advantage of better edge preservation with higher order filters, too.

As generalisation of this methods, we have taken a look at variational methods whose data terms involve the function and its derivatives. These methods seem to be promising by leading to a better adaptation of the solution to important image features. They can improve the denoising quality compared to fourth order regularised total variation filtering.

Chapter 3

Wavelet Shrinkage and Integrodifferential Equations

In the last chapter, the focus was on derivative-based regularisation methods and general diffusion-type evolution equations. This chapter now gives relations between similar derivative-based approaches and wavelet shrinkage.

So far, the relations between wavelet shrinkage and nonlinear diffusion equations have been fairly well-understood in the single-scale case. We are going to extend these results to the practically relevant multiscale setting here. Multiscale wavelet shrinkage is related to novel integrodifferential equations which differ from nonlinear diffusion by the fact that all appearing derivatives are smoothed with convolution kernels. An integration over all scales makes sure that there is no preferred scale during filtering. Furthermore, the convolution-based smoothed derivative operators can be expressed as pseudodifferential operators by power series of differential operators. This makes a formulation without convolutions possible.

In the first section, we consider wavelet shrinkage on one-dimensional functions. This allows us to consider arbitrary translations and dilations of the wavelets. The discrete variant working on signals and two-dimensional images is discussed in Section 3.2. Most of the results in the first section can be reformulated in the discrete setting. However, the scaling operations on kernels require a slightly more technical background here. Section 3.3 gives a summary and concludes this chapter.

3.1 Relations in the Continuous Setting

In this section, we consider wavelet shrinkage in the practically relevant *multiscale* setting. For the sake of simplicity, we focus on one-dimensional signals and we analyse the continuous shrinkage framework first. The key observation exploited here is the fact that, for wavelets with a finite number of vanishing moments, the

wavelet transform can be understood as applying a smoothed derivative operator [128, p. 167]. Thus, wavelet shrinkage on a single scale is closely related to a diffusion type equation where all appearing spatial derivatives are regularised with a convolution kernel. Going from a single scale to multiple scales introduces a further integration, yielding a novel integrodifferential equation derived from wavelet shrinkage. Moreover, we express convolution with a smoothing kernel by a power series of differential operators. This allows us to regard multiscale wavelet shrinkage as an averaging of pseudodifferential equations over a continuum of scales. These results make the analytical reasons for the differences between continuous multiscale wavelet shrinkage on one side and nonlinear diffusion and its corresponding variational regularisation on the other side explicit: They are caused by the presence of additional integration scales, smoothing operators and differential operators of higher order.

This section is organised as follows: Subsection 3.1.1 introduces some useful notations and summarises the classical continuous wavelet shrinkage approach as we will need it here. The idea of understanding wavelets as smoothed derivative operators, which is crucial for the remainder of this whole chapter, is explained in Subsection 3.1.2. With this knowledge, Subsection 3.1.3 describes how wavelet shrinkage can be interpreted as approximation to a novel integrodifferential evolution equation. In Subsection 3.1.4 we are going to present a corresponding energy functional that uses both smoothed derivative operators within the penaliser and integration over all scales. The link to pseudodifferential operators is discussed in Subsection 3.1.5.

3.1.1 Wavelet Shrinkage

This subsection makes the ideas behind wavelet shrinkage already sketched in Section 1.3 precise in a suitable way for our further considerations.

Before we formulate continuous wavelet shrinkage, let us give an introduction to the formal notions related to wavelets which will be used here. In this and the following three sections, we consider signals as real functions $f, u \in \mathscr{L}^1(\mathbb{R})$. We choose a real function $\psi \in \mathscr{L}^1(\mathbb{R}) \cap \mathscr{L}^2(\mathbb{R})$, the *mother wavelet*, which has to satisfy the *admissibility condition* [52, p. 27]

$$c_\psi \;:=\; 2\pi \int_0^\infty \frac{|\hat{\psi}(\xi)|^2}{\xi}\, d\xi \;<\; \infty \;. \tag{3.1}$$

Here, the Fourier transform is defined as

$$\hat{\psi}(\xi) \;=\; \mathcal{F}\psi(\xi) \;:=\; \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \psi(x) \exp(-ix\xi)\, dx \;.$$

To simplify the notation let ψ_σ and $\tilde{\psi}$ be scaled and mirrored versions of ψ,

respectively, i. e.

$$\psi_\sigma(x) := \frac{1}{\sqrt{c_\psi \sigma}} \psi\left(\frac{x}{\sigma}\right) \quad \text{and} \quad \tilde{\psi}(x) := \psi(-x) \tag{3.2}$$

for $\sigma > 0$. By $f * g$ we denote the convolution of two functions $f, g \in \mathscr{L}^1(\mathbb{R})$:

$$(f * g)(x) \;=\; \int_{-\infty}^{\infty} f(x-\tau) g(\tau)\, d\tau \quad \text{for all } x \in \mathbb{R} \;. \tag{3.3}$$

If a function $f(x, y)$ depends on more than one variable we replace the variable in which the convolution is performed with a dot, for example

$$\left(f(\cdot, y) * g\right)(x) \;=\; \int_{-\infty}^{\infty} f(x-\tau, y) g(\tau)\, d\tau \quad \text{for all } x, y \in \mathbb{R} \;. \tag{3.4}$$

This notations help us to write down wavelet shrinkage [194, 65] in an easy way. Wavelet shrinkage transforms the data in a suitable representation and performs simple nonlinear operations. The back-transform finally yields the denoising result. Let us now formulate these three steps in detail:

1. **Analysis:** First the given function f is transformed into the wavelet domain. With the notations introduced above we can write the wavelet transform as

   $$\mathcal{W}_\psi f(x, \sigma) \;:=\; \frac{1}{\sqrt{c_\psi}} \int_{-\infty}^{\infty} f(\tau) \frac{1}{\sqrt{\sigma}} \psi\left(\frac{\tau - x}{\sigma}\right) d\tau \;=\; \left(\tilde{\psi}_\sigma * f\right)(x) \;. \tag{3.5}$$

   Thus this step can be seen as convolution with scaled and mirrored versions of the mother wavelet.

2. **Shrinkage:** A – typically nonlinear – shrinkage function $S : \mathbb{R} \longrightarrow \mathbb{R}$ is applied to the wavelet transform $\mathcal{W}_\psi f$. Usually it is assumed that this shrinkage function diminishes the absolute value of the wavelet coefficients without changing their sign. Reasonable assumptions on S are thus

   $$x > 0 \implies S(x) \geq 0 \;, \quad S(-x) = -S(x) \;, \quad \text{and } |S(x)| \leq |x| \tag{3.6}$$

   for all $x \in \mathbb{R}$. Usually S depends on a parameter λ which determines the amount of shrinkage. This parameter is omitted here to simplify the notations. Examples of typical shrinkage functions can be found in Table 3.1.

3. **Synthesis:** As final step, the shrunken wavelet transform $S \circ \mathcal{W}_\psi f$ has to be transformed back into the spatial domain to yield the resulting function u of the shrinkage. It should be mentioned that the wavelet transform $\mathcal{W}$

is an isometric mapping from $\mathscr{L}^2(\mathbb{R})$ to $\mathscr{L}^2(\mathbb{R} \times \mathbb{R}_0^+)$, equipped with the inner product

$$\langle f, g \rangle \;:=\; \int_{-\infty}^{\infty} \int_0^{\infty} f(x,\sigma)\, \bar{g}(x,\sigma)\, \frac{d\sigma}{\sigma^2}\, dx \tag{3.7}$$

where $\bar{g}$ is the complex conjugate of g. Thus, on the subspace given by the image of $\mathcal{W}_\psi$, the adjoint operator $\mathcal{W}_\psi^*$ is the inverse of $\mathcal{W}_\psi$. We show several ways to formulate the back-transform at the example of the shrunken coefficients:

$$\begin{aligned} u &= \mathcal{W}_\psi^*(S \circ \mathcal{W}_\psi f) \\ &= \int_0^{\infty} \Big(\psi_\sigma * S(\mathcal{W}_\psi f(\cdot,\sigma))\Big)\, \frac{d\sigma}{\sigma^2} \\ &= \frac{1}{\sqrt{c_\psi}} \int_{-\infty}^{\infty} \int_0^{\infty} S(\mathcal{W}_\psi f(\tau,\sigma))\, \frac{1}{\sqrt{\sigma}}\, \psi\left(\frac{\cdot - \tau}{\sigma}\right)\, \frac{d\sigma}{\sigma^2}\, d\tau \;. \end{aligned} \tag{3.8}$$

The convergence of these integrals should be understood in the weak sense, see [52, p. 25]. In the following sections we will mostly refer to the following formulation:

$$u \;=\; \int_0^{\infty} \Big(\psi_\sigma * S(\tilde{\psi}_\sigma * f)\Big)\, \frac{d\sigma}{\sigma^2} \;. \tag{3.9}$$

Note that besides the convolution with ψ_σ, the back-transform also introduces an integration over all scales σ.

This formulation for wavelet shrinkage will be the starting point for our considerations in the following.

3.1.2 Wavelet Transforms as Smoothed Derivative Operators

In Section 3.1.1 we have introduced the wavelet shrinkage technique with an arbitrary mother wavelet ψ. Now we restrict our choice to a certain class of wavelets in order to relate the corresponding wavelet transform to smoothed derivative operators.

First we assume that the mother wavelet ψ has fast decay, i. e. for any exponent $m \in \mathbb{N}$ there exists a constant c_m such that

$$|\psi(x)| \;\leq\; \frac{c_m}{1 + |x|^m} \quad \text{for all } x \in \mathbb{R} \;. \tag{3.10}$$

For the rest of this section, we focus on wavelets with a finite number $p \in \mathbb{N} \setminus \{0\}$ of vanishing moments:

$$\int_{-\infty}^{\infty} x^k \psi(x)\, dx = 0 \quad \text{for } 0 \leq k < p \;. \tag{3.11}$$

It is well-known [128, p. 167] that these assumptions are equivalent to the existence of a function θ with fast decay such that

$$\psi(x) \;=\; (-1)^p \, \frac{d^p}{dx^p} \, \theta(x) \;\; . \tag{3.12}$$

Moreover, ψ has no more than p vanishing moments if and only if θ has nonzero mean value, i. e. $\int_{-\infty}^{\infty} \theta(x) \; dx \neq 0$. In our further considerations, this function θ will play the role of a smoothing kernel. Keeping this in mind it makes sense to consider the maximal number of vanishing moments for a certain wavelet which gives a natural choice of the corresponding derivative order.

Example 3.1 (Wavelets and Smoothing Kernels)
Figure 3.1 gives two examples of wavelets that are often used in image processing together with their corresponding derivative orders and smoothing kernels. It can be seen that the Haar wavelet is the first derivative of a hat-shaped function. We see that this hat function has a negative sign which is not typical for a convolution kernel. Since we only use the absolute value of the convolution result or use the convolution and its adjoint operator after each other, this negative sign does not change the results in our applications. The above cited theorem does not make a statement about the sign of the kernels, and so we can not expect it to be positive. The second classical example is the Mexican hat wavelet which is defined as the second derivative of a negative Gaussian kernel. Examples of wavelets with a higher number of vanishing moments include the Daubechies wavelets [52]. All the wavelet classes mentioned here are also of fast decay, the Mexican hat wavelet because of its exponential decreasing velocity, and the others because of their compact support. We are going to see more examples for wavelets and their corresponding smoothing kernels in the next section.

So far, we have seen the relation (3.12) between the mother wavelet and the smoothing kernel. It is not difficult to verify the following equations for scaled and mirrored versions of the wavelet:

$$\tilde{\psi}_\sigma * f \;=\; \sigma^p \, \partial_x^p \left(\tilde{\theta}_\sigma * f \right) \;=\; \sigma^p \, (\tilde{\theta}_\sigma)^{(p)} * f \tag{3.13}$$

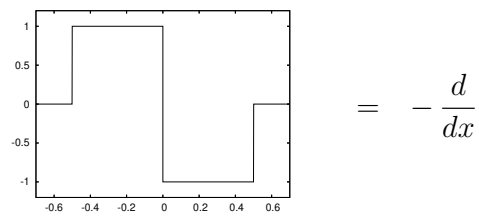
and

$$\psi_\sigma * f \;=\; (-\sigma)^p \, \partial_x^p \, (\theta_\sigma * f) \;=\; (-\sigma)^p \, (\theta_\sigma)^{(p)} * f \;\; . \tag{3.14}$$

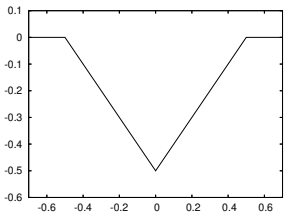
As an elementary property of the convolution, one can also put the derivative in front of the function f, if the regularity of f allows for this. If this is not the case one can motivate the smoothing with the kernel θ as regularisation of the p-th derivative of f.

Equation (3.13) shows that the wavelet transform is equivalent to taking a smoothed derivative with an additional weight factor σ^p:

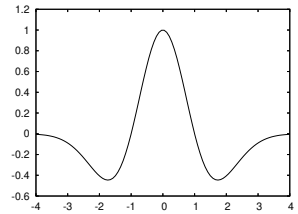
$$\mathcal{W}_\psi f \;=\; \left(\tilde{\psi}_\sigma * f \right) \;=\; \sigma^p \, \partial_x^p \left(\tilde{\theta}_\sigma * f \right) \;\; . \tag{3.15}$$



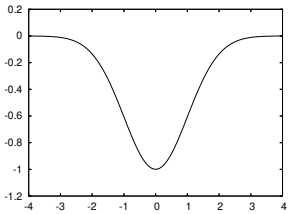
$$= \quad -\frac{d}{dx}$$



Haar wavelet [96] (negative) hat function



$$= \quad \frac{d^2}{dx^2}$$



Mexican hat wavelet [128] (negative) Gaussian kernel

Figure 3.1: Examples of wavelets as derivatives of smoothing kernels.

For the back-transform, an additional integration over all scales σ is introduced:

$$\mathcal{W}_\psi^* f \;=\; \int_0^\infty (\psi_\sigma * f)\;\frac{d\sigma}{\sigma^2} \;=\; \int_0^\infty (-\sigma)^p\,\partial_x^p\,(\theta_\sigma * f)\;\frac{d\sigma}{\sigma^2}\;. \tag{3.16}$$

These two equations will form the basis for relating wavelet techniques to derivative-based methods.

3.1.3 Wavelet Shrinkage and Evolution Equations

In this section, we are going to relate wavelet shrinkage as given in Subsection 3.1.1 to novel integrodifferential evolution equations involving the data on a continuous spectrum of scales.

We remember the prototype (1.17) of an evolution equation:

$$\partial_t u \;=\; \mathcal{L}^*\Big(g\left(|\mathcal{L}u|^2\right)\mathcal{L}u\Big)\;. \tag{3.17}$$

In principle we will study this in the case that $\mathcal{L} = \mathcal{W}_\psi$ is the wavelet transform. Additionally we are going to argue that the time-discrete shrinkage steps can be seen just as an approximation to a time-continuous evolution. The ideas we follow here are adapted from [140] where similar relations have been considered in the discrete setting on the finest scale.

We start with writing down one step of wavelet shrinkage as given in Section 3.1.1:

$$u \;=\; \int_0^\infty \Big(\psi_\sigma * S(\tilde{\psi}_\sigma * f)\Big)\;\frac{d\sigma}{\sigma^2}\;. \tag{3.18}$$

Now we choose a function g such that

$$S(x) \;=\; x - \tau g(|x|^2)\, x \;\;, \tag{3.19}$$

where $\tau > 0$ has the meaning of a time step size. During the further steps we are going to see immediately why this choice makes sense. The function g is going to play the role of a diffusivity function in the following considerations. Under the assumptions (3.6) it follows that $0 \leq g(|x|^2) \leq 1/\tau$. This range is in accordance with stability requirements that do not allow negative diffusivities. We have seen in the last chapter that limiting the product of diffusivity and time step size from above by 1 is a very helpful normalisation in image analysis.

Plugging (3.19) into the wavelet shrinkage formula (3.18) yields

$$\begin{aligned} u \;&=\; \int_0^\infty \Big(\psi_\sigma * \tilde{\psi}_\sigma * f\Big)\; \frac{d\sigma}{\sigma^2} \\ &\quad - \tau \int_0^\infty \Big(\psi_\sigma * \Big(g(|\tilde{\psi}_\sigma * f|^2)(\tilde{\psi}_\sigma * f)\Big)\Big)\; \frac{d\sigma}{\sigma^2} \;\;. \end{aligned} \tag{3.20}$$

The first integral is simply the wavelet reconstruction formula which results in the initial data f. Therefore this expression is equivalent to

$$\frac{u-f}{\tau} \;=\; -\int_0^\infty \Big(\psi_\sigma * \Big(g(|\tilde{\psi}_\sigma * f|^2)\,(\tilde{\psi}_\sigma * f)\Big)\Big)\; \frac{d\sigma}{\sigma^2} \;\;. \tag{3.21}$$

Now the next step is to involve the ideas from Subsection 3.1.2 to regard wavelets as smoothed derivative operators. Using (3.15) and (3.16) we obtain the following equivalent formulation of wavelet shrinkage as integrodifferential equation:

$$\frac{u-f}{\tau} \;=\; (-1)^{p+1} \int_0^\infty \sigma^{2p}\, \partial_x^p\, \theta_\sigma * \Big(g(|\sigma^p\, \partial_x^p\, \tilde{\theta}_\sigma * f|^2)(\partial_x^p\, \tilde{\theta}_\sigma * f)\Big)\; \frac{d\sigma}{\sigma^2} \;\;. \tag{3.22}$$

Let us now relate this process to nonlinear diffusion filtering. Similar to the approach already sketched in Subsection 1.2.2 we now introduce an artificial time variable t for the function u such that the initial data is given at time $t = 0$: $u(\cdot, 0) = f$. It is possible to write the left-hand side as $\frac{u(x)-f(x)}{\tau} = \frac{u(x,\tau)-u(x,0)}{\tau}$ with time step size τ. Keeping this in mind, (3.22) can be understood as a single step time-explicit approximation to the evolution equation

$$\partial_t u \;=\; (-1)^{p+1} \int_0^\infty \sigma^{2p}\, \partial_x^p\, \theta_\sigma * \Big(g(|\sigma^p\, \partial_x^p\, \tilde{\theta}_\sigma * u|^2)(\partial_x^p\, \tilde{\theta}_\sigma * u)\Big)\; \frac{d\sigma}{\sigma^2} \;\;. \tag{3.23}$$

Interpretation. If we compare (3.23) to the higher order nonlinear diffusion equation (2.36) introduced as Example 2.18

$$\partial_t u \;=\; (-1)^{p+1}\, \partial_x^p \Big(g\left(|\partial_x^p u|^2\right) \partial_x^p u\Big) \;\;,$$

we notice two differences:

1. All appearing derivative operators in the equation are presmoothed by convolution with scaled and mirrored versions of a kernel θ.

2. The right-hand side is not only considered at one single scale, but there is an integration over all scales with additional weight factors.

We are going to see in the next section that these two steps can also be used to turn classical variational methods into wavelet-based ones.

Let us now take a closer look at the relation between shrinkage functions and diffusivities introduced by (3.19):

Remark 3.2 (Shrinkage Functions and Diffusivities)
Table 3.1 shows some corresponding pairs of shrinkage functions S and diffusivities g. We observe that several practically used shrinkage functions lead to well-known diffusivities. An interesting case is for example soft wavelet shrinkage [194] that corresponds via (3.19) to a total variation (TV) diffusivity [4] which is regularised for small parameters where it would become unbounded. A similar regularisation for small parameters appears in the balanced-forward-backward (BFB) diffusivity [109] corresponding to garrote shrinkage [85]. With the Perona-Malik diffusivity [154] and the diffusivity related to the regularisation approach of Charbonnier et al. [39] we also give two examples where we start with classical diffusivities and calculate corresponding shrinkage functions.

We have seen how wavelet shrinkage can be related to integrodifferential equations. In the next section we consider regularisation approaches and investigate how these can be related to variational formulations for wavelet shrinkage.

3.1.4 Variational Methods and Correspondences

Instead of starting with the classical wavelet shrinkage approach, we now adapt the variational formulation to extend our understanding of the similarities. Such variational formulations using wavelets have been studied in [34, 35, 14], for example. Usually they work with orthogonal wavelet representations which has the advantage that the wavelet coefficients are independent: The minimisation problems come down to minimisation of real functions.

Let us formulate an approach for wavelet shrinkage using the prototype for variational image simplification methods (1.14). Here we consider the case that $\mathcal{L}$ is the wavelet transform as introduced in Subsection 3.1.1, so we have $\Gamma = \mathbb{R} \times \mathbb{R}_0^+$ and integrate with the corresponding weight. Thus we formulate an energy functional for wavelet shrinkage as

$$\mathscr{E}(u) \;=\; \int_{-\infty}^{\infty} (u-f)^2 \, dx \;+\; \alpha \int_{-\infty}^{\infty} \int_{0}^{\infty} \Psi\left(|\mathcal{W}_\psi u(x,\sigma)|^2\right) \; \frac{d\sigma}{\sigma^2} \, dx \;. \qquad (3.24)$$

shrinkage	$S(x)$	$g(\|x\|^2)$	diffusivity
linear	λx	$\dfrac{1-\lambda}{\tau}$	Linear [102]
hard [128]	$\begin{array}{ll} 0, & \|x\| \le \lambda \\ x, & \|x\| > \lambda \end{array}$	$\begin{array}{ll} \frac{1}{\tau}, & \|x\| \le \lambda \\ 0, & \|x\| > \lambda \end{array}$	
soft [194]	$\begin{array}{ll} 0, & \|x\| \le \lambda \\ x - \lambda \operatorname{sgn}(x), & \|x\| > \lambda \end{array}$	$\begin{array}{ll} \frac{1}{\tau}, & \|x\| \le \lambda \\ \frac{\lambda}{\tau\|x\|}, & \|x\| > \lambda \end{array}$	$\approx$ TV [4, 158]
garrote [85]	$\begin{array}{ll} 0, & \|x\| \le \lambda \\ x - \frac{\lambda^2}{x}, & \|x\| > \lambda \end{array}$	$\begin{array}{ll} \frac{1}{\tau}, & \|x\| \le \lambda \\ \frac{\lambda^2}{\tau x^2}, & \|x\| > \lambda \end{array}$	$\approx$ BFB [109]
	$\dfrac{x^3}{\lambda^2 + x^2}$	$\dfrac{1}{\tau}\left(1 + \dfrac{x^2}{\lambda^2}\right)^{-1}$	Perona-Malik [154]
	$\left(1 - \sqrt{\dfrac{\lambda^2}{\lambda^2 + x^2}}\right) x$	$\dfrac{1}{\tau}\left(1 + \dfrac{x^2}{\lambda^2}\right)^{-\frac{1}{2}}$	Charbonnier [39]

Table 3.1: Shrinkage functions and corresponding diffusivities. Adapted from [140].

Replacing the wavelet transform operator by its convolution formulation, this functional reads as

$$\mathscr{E}(u) \;=\; \int_{-\infty}^{\infty} (u-f)^2 \, dx \;+\; \alpha \int_{-\infty}^{\infty} \int_{0}^{\infty} \Psi\left(|\tilde{\psi}_\sigma * u|^2\right) \, \frac{d\sigma}{\sigma^2} \, dx \;. \tag{3.25}$$

The difference to the variational formulations presented in [34, 35, 14] is that we have not sampled the scales and translations in the wavelet domain, but stay in the continuous setting.

With (3.13) we can directly rewrite this energy functional as

$$\mathscr{E}(u) \;=\; \int_{-\infty}^{\infty} (u-f)^2 \, dx \;+\; \alpha \int_{-\infty}^{\infty} \int_{0}^{\infty} \Psi\left(|\sigma^p \, \partial_x^p (\tilde{\theta}_\sigma * u)|^2\right) \, \frac{d\sigma}{\sigma^2} \, dx \tag{3.26}$$

which shows that we are indeed penalising a presmoothed derivative of our image u. Again we see that the same two steps as in the last section lead to classical regularisation approaches (1.3). First we leave out the integration over all scales and obtain a variational functional

$$\mathscr{E}(u) \;=\; \int_{-\infty}^{\infty} (u-f)^2 \, dx \;+\; \alpha \int_{-\infty}^{\infty} \Psi\left(|\partial_x^p \, (\tilde{\theta}_\sigma * u)|^2\right) \, dx \;. \tag{3.27}$$

A special case ($n = 1$) of this functional is given by (1.4) which has been considered by Scherzer and Weickert [165]. In our case a richer choice of convolution kernels and higher derivative orders is allowed. The second step is to omit also the convolution. This leads us directly to the classical regularisation functionals (1.3).

By comparing these results with the last section, we see that the two major differences between multiscale wavelet shrinkage and regularisation are the same as between multiscale wavelet shrinkage and nonlinear diffusion: smoothed derivative operators instead of derivatives, and weighted integration over all scales instead of working at the finest scale.

3.1.5 Smoothing Kernels and Pseudodifferential Operators

So far, we have related wavelet shrinkage to integrodifferential equations which involve convolutions with scaled and mirrored versions of the mother wavelet. In this section, we express these convolutions as pseudodifferential operators. This allows us to eliminate all integrals from (3.23) except the integration over all scales. Instead of the convolutions, the equations then contain power series of differential operators.

After introducing some notations and technical details, we are going to describe the general procedure. Then we apply this to examples for convolution kernels, namely to a box function and a Gaussian kernel. As we have seen in Subection 3.1.1, also the wavelet transform is a convolution and thus considered after the first two examples. Since the Haar and the Mexican hat wavelet can be written as derivatives of a hat function and a Gaussian, we can conclude the section with these two popular examples of wavelets.

Let $\mathscr{S}(\mathbb{R})$ be the Schwartz space of rapidly decreasing functions [181, 182]. We consider the convolution $\theta * f$ of a function $f \in \mathscr{S}(\mathbb{R})$ with a kernel $\theta \in \mathscr{L}^1(\mathbb{R}) \cap \mathscr{L}^2(\mathbb{R})$. It is well-known that convolution in the spatial domain is equivalent to multiplication in the Fourier domain:

$$\mathcal{F}(h * f) \;=\; \sqrt{2\pi}\; \hat{h} \cdot \hat{f} \;\; . \tag{3.28}$$

Besides convolutions, also derivative operators are multiplications in the Fourier domain, namely

$$\mathcal{F}\left(\frac{d}{dx} f\right) \;=\; i\xi \hat{f} \;\; . \tag{3.29}$$

Let us assume that the Fourier transform of our convolution kernel θ is analytic, i. e. there is a power series representation

$$\hat{\theta}(\xi) \;=\; \sum_{k=0}^{\infty} a_k \, \xi^k \;\; . \tag{3.30}$$

For $f \in \mathscr{S}(\mathbb{R})$ we can understand the product $\hat{\theta} \cdot \hat{f}$ as a sum of derivatives of f up to arbitrary orders in the Fourier domain:

$$\frac{1}{\sqrt{2\pi}}\, \theta * f \;=\; \mathcal{F}^{-1}(\hat{\theta} \cdot \hat{f}) \;=\; \mathcal{F}^{-1}\left(\sum_{k=0}^{\infty} a_k\, \xi^k\, \hat{f} \right) \;. \tag{3.31}$$

Under the assumption of sufficient convergence conditions of the power series, we may interchange the sum and the Fourier back-transform which allows us to write

$$\mathcal{F}^{-1}\left(\sum_{k=0}^{\infty} a_k\, \xi^k\, \hat{f} \right) \;=\; \sum_{k=0}^{\infty} a_k\, \mathcal{F}^{-1}(\xi^k \hat{f}) \;=\; \sum_{k=0}^{\infty} a_k \left(\frac{1}{i}\, \frac{d}{dx} \right)^k f \;. \tag{3.32}$$

In this context the symbol $\hat{\theta}\left(\frac{1}{i}\frac{d}{dx}\right)$ is used to denote this power series of differential operators (see [182], for example):

$$\hat{\theta}\left(\frac{1}{i}\, \frac{d}{dx} \right) f \;:=\; \sum_{k=0}^{\infty} a_k \left(\frac{1}{i}\, \frac{d}{dx} \right)^k f \;=\; \frac{1}{\sqrt{2\pi}}\, \theta * f \;. \tag{3.33}$$

We reformulate this as the central equation

$$\theta * f \;=\; \sqrt{2\pi}\; \hat{\theta}\left(\frac{1}{i}\, \frac{d}{dx} \right) f \tag{3.34}$$

relating convolution to power series of derivatives. It is also well-known that such a reasoning can be generalised from analytic functions to richer classes of functions $\hat{\theta}$, for example continuous or measurable functions [181, 209].

After describing the general idea, let us now apply this to two examples of convolution kernels:

Example 3.3 (Convolution with a Box Function)
A basic operation in image processing is to take the arithmetic mean inside a symmetric neighbourhood of pixels. This can be understood as convolution with the characteristic function of an interval

$$\chi(x) \;:=\; \chi_{[-\frac{1}{2},\frac{1}{2}]}(x) \;=\; \begin{cases} 1, & x \in [-\frac{1}{2}, \frac{1}{2}] \\ 0, & \text{else} \;. \end{cases} \tag{3.35}$$

This function has the Fourier transform

$$\hat{\chi}(\xi) = \frac{1}{\sqrt{2\pi}}\, \mathrm{sinc}\left(\frac{1}{2}\xi \right) = \frac{1}{\sqrt{2\pi}}\, \frac{\sin\left(\frac{1}{2}\xi\right)}{\frac{1}{2}\xi} = \frac{1}{\sqrt{2\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k \xi^{2k}}{4^k (2k+1)!} \;. \tag{3.36}$$

Thus we write the convolution with the box function χ as pseudodifferential operator

$$\chi * f \;=\; \mathrm{sinc}\left(\frac{1}{2i}\, \frac{d}{dx} \right) f \;=\; \sum_{k=0}^{\infty} \frac{1}{4^k (2k+1)!}\, \frac{d^{2k}}{dx^{2k}}\, f \;. \tag{3.37}$$

Example 3.4 (Convolution with a Gaussian Kernel)
As our second example, we take a look at the convolution with a Gaussian kernel, since this operation is fundamental in image processing. It is well-known [182] that the Fourier transform of a Gaussian kernel $\theta(x) = \exp\left(-\frac{x^2}{2}\right)$ is again a Gaussian function $\hat{\theta}(\xi) = \exp\left(-\frac{\xi^2}{2}\right)$. With the above reasoning we then write

$$\theta * f \;=\; \sqrt{2\pi}\,\exp\left(\frac{1}{2}\frac{d^2}{dx^2}\right) f \;=\; \sqrt{2\pi}\sum_{k=0}^{\infty}\frac{1}{2^k k!}\,\frac{d^{2k}}{dx^{2k}}\,f \;. \tag{3.38}$$

Writing down the first summands of this power series explicitly gives

$$\theta * f \;=\; \sqrt{2\pi}\left(f + \frac{1}{2}\,\frac{d^2 f}{dx^2} + \frac{1}{8}\,\frac{d^4 f}{dx^4} + \frac{1}{48}\,\frac{d^6 f}{dx^6} + \dots\right) \tag{3.39}$$

Similar formulations often appear in the context of linear scale-spaces and the heat equation (see [143, 69, 182]). For example, Nielsen et al. [143] proposed to express the Gaussian scale-space via minimisation of an energy functional with a power series of derivatives in the smoothness term.

After these two examples, we now apply the idea to the convolution coming from the wavelet transform.

General Wavelet Transform: We have seen in Subsection 3.1.1 how the wavelet transform can be understood as convolution operator. Given a mother wavelet ψ, the Riemann-Lebesgue theorem assures that its Fourier transform is integrable, i. e. $\hat{\psi} \in \mathscr{L}^1(\mathbb{R})$. This allows us to use (3.34) and write

$$\psi * f \;=\; \sqrt{2\pi}\;\hat{\psi}\left(\frac{1}{i}\,\frac{d}{dx}\right) f \;. \tag{3.40}$$

For a wavelet transform we now need convolutions with translated and scaled versions of the mother wavelet ψ which can be obtained in a general way. The translation is simply the evaluation of $\psi * f$ at another point. For the scaled and mirrored version a substitution shows that

$$(\mathcal{F}\tilde{\psi}_\sigma)(\xi) \;=\; -\sqrt{\frac{\sigma}{c_\psi}}\,\hat{\psi}(-\sigma\xi) \;. \tag{3.41}$$

Together this means that we can write the wavelet transform as

$$\mathcal{W}_\psi f(x,\sigma) \;=\; \tilde{\psi}_\sigma * f(x) \;=\; -\sqrt{\frac{2\pi\sigma}{c_\psi}}\left(\hat{\psi}\left(-\frac{\sigma}{i}\frac{d}{dx}\right) f\right)(x) \tag{3.42}$$

and thus express it as pseudodifferential operator.

We will make this more explicit with the two examples of Haar and Mexican hat wavelets:

Example 3.5 (Haar Wavelet)
Here, we consider a shifted version of the Haar wavelet defined as

$$\psi(x) \;=\; \begin{cases} 1\ , & -\frac{1}{2} \le x \le 0 \\ -1\ , & 0 < x \le \frac{1}{2} \\ 0\ , & \text{else}\ . \end{cases} \tag{3.43}$$

This shift is not important since we only consider convolutions with this function, and convolutions are shift invariant. The Haar wavelet is the first derivative of a hat function

$$h(x) \;=\; \begin{cases} \frac{1}{2} - |x|\ , & |x| \le \frac{1}{2} \\ 0\ , & \text{else}\ . \end{cases} \tag{3.44}$$

A simple calculation shows that h can be written in terms of the box function χ considered above, namely $h = \chi(2\cdot) * \chi(2\cdot)$. Together we have

$$\psi \;=\; \frac{d}{dx}(\chi(2\cdot) * \chi(2\cdot))\ . \tag{3.45}$$

With (3.36) this implies in the Fourier domain

$$\hat{\psi}(\xi) \;=\; \frac{i\xi}{4\sqrt{2\pi}}\,\mathrm{sinc}^2\left(\frac{\xi}{4}\right)\ . \tag{3.46}$$

Written as pseudodifferential operator the basic convolution of a Haar wavelet transform looks as follows:

$$\psi * f \;=\; \frac{1}{4}\frac{d}{dx}\,\mathrm{sinc}^2\left(\frac{1}{4i}\frac{d}{dx}\right) f\ . \tag{3.47}$$

For the scaled version of ψ we can use (3.42). The power series looks as follows:

$$\tilde{\psi}_\sigma * f \;=\; -\frac{1}{4}\sqrt{\frac{\sigma}{c_\psi}}\,\frac{d}{dx}\,\mathrm{sinc}^2\left(\frac{-\sigma}{4i}\frac{d}{dx}\right) f\ . \tag{3.48}$$

With this equation we have expressed the wavelet transform with a Haar wavelet at scale σ. The convolutions appearing in the back-transform are the same except for the mirroring of the kernel.

Example 3.6 (Mexican Hat Wavelet)
The Mexican hat mother wavelet is given by the second derivative of a Gaussian function with negative sign

$$\psi(x) \;=\; -\frac{d^2}{dx^2}\exp\left(-\frac{x^2}{2}\right)\ .$$

The Gaussian has already been considered in (3.38), so that we can directly give the corresponding derivative operator

$$\psi * f \;=\; -\sqrt{2\pi}\,\frac{d^2}{dx^2}\,\exp\left(\frac{1}{2}\,\frac{d^2}{dx^2}\right) f \;=\; -\sqrt{2\pi}\sum_{k=0}^{\infty}\frac{1}{2^k k!}\,\frac{d^{2k+2}}{dx^{2k+2}}\,f\ . \tag{3.49}$$

Again we consider scaled and mirrored versions since they appear as convolution kernels in the wavelet transform:

$$\tilde{\psi}_\sigma * f \;=\; \frac{\sqrt{2\pi\sigma}}{c_\psi}\,\frac{d^2}{dx^2}\,\exp\left(\frac{\sigma^2}{2}\frac{d^2}{dx^2}\right) f \;=\; \sqrt{2\pi\sigma}\sum_{k=0}^{\infty}\frac{\sigma^{2k}}{2^k k!}\,\frac{d^{2k+2}}{dx^{2k+2}}\,f \;. \qquad (3.50)$$

We have used here that $c_\psi = 1$ for the mexican hat (see [120], for example). To get an intuition of what happens here, we write the first summands of this sum out and obtain:

$$\tilde{\psi}_\sigma * f \;=\; \sqrt{2\pi\sigma}\left(\frac{d^2 f}{dx^2} + \frac{\sigma^2}{2}\,\frac{d^4 f}{dx^4} + \frac{\sigma^4}{8}\,\frac{d^6 f}{dx^6} + \frac{\sigma^6}{48}\,\frac{d^8 f}{dx^8} + \ldots\right) \;. \qquad (3.51)$$

These examples show that it is possible to reformulate the convolutions in the integrodifferential evolution equation (3.23) as power series of differential operators. This eliminates all integrations in this equation except for one: The outer integral over all scales σ still remains.

From this point of view, the PDE methods described in Chapter 2 use very crude approximations of these power series by taking only the first summand and considering only one scale. In the previous example, this would lead to a fourth order diffusion equation, since the first summand in (3.51) is the second derivative of f.

3.2 Relations in the Discrete Setting

So far, we have studied wavelet shrinkage and corresponding integrodifferential equations in the continuous one-dimensional setting. Let us now transfer the results from the last section from the continuous to the discrete setting. Since the dilation operation on the wavelets can only be approximated on a discrete pixel grid, the formulation is a bit more technical here. Moreover, we will not restrict ourselves to orthogonal wavelets, but also take a look at biorthogonal ones allowing for more general integrodifferential equations. We are going to transfer the one-dimensional case to two dimensions using tensor product wavelets and special shrinkage rules to increase rotational invariance. Numerical experiments will be shown to compare the resulting methods in terms of denoising quality.

3.2.1 Description of the Classical Methods

In this section, we define some necessary notations and review the classical methods of wavelet shrinkage and higher order nonlinear diffusion in the case of one-dimensional discrete signals.

Preliminaries and Notations

Definition 3.7 *Let* $f \in \ell^2(\mathbb{Z}) := \{(f_n)_{n\in\mathbb{Z}} \mid \sum_{n=-\infty}^{\infty} f_n^2 < \infty\}$ *be a signal. Then*

$$\hat{f}(\omega) := \sum_{n=-\infty}^{\infty} f_n \exp(-in\xi) \quad \textit{and} \quad F(z) := \sum_{n=-\infty}^{\infty} f_n z^{-n} \tag{3.52}$$

denote the Fourier- and the z-transform of f*.*

The importance of both transforms in this context mainly results from the fact that they allow for an easy way to express convolutions as multiplications of formal (trigonometric) Laurent series. Since we work with signals and filters of finite length in practice, these series are in fact just finite sums or polynomials (up to a shift by multiplication with a monomial $\exp(ij\omega)$ or z^j). In the case of signals with finite length, we are going to assume periodic boundary conditions or periodic extension of the signals. For the z-transforms, we have to consider the exponents modulo the number N of pixels in this case. In this setting, convolutions are equivalent to circulant matrices (see [100], for example).

Definition 3.8 (Circulant Matrix)
Given a vector $a \in \mathbb{R}^N$*, the corresponding circulant matrix is defined as*

$$A := \begin{pmatrix} a_0 & a_1 & a_2 & \dots & a_{N-1} \\ a_{N-1} & a_0 & a_1 & \dots & a_{N-2} \\ a_{N-2} & a_{N-1} & a_0 & \dots & a_{N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & a_3 & \dots & a_0 \end{pmatrix} \in \mathbb{R}^{N\times N} .$$

Multiplying such a matrix with a vector f implements the discrete convolution $a * f$ with periodic boundary conditions. Each circulant matrix can be written as

$$A = \sum_{j=0}^{N-1} a_j C^j \quad \text{where} \quad C := \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \end{pmatrix}$$

is the so-called *basic circulant permutation matrix*. Multiplication with C performs a periodic left-shift of a vector.

In the following we will often use some vector $a \in \mathbb{R}^N$ in connection with its corresponding N-dimensional circulant matrix $A = \sum_{j=0}^{N-1} a_j C^j$ and its z-transform $A(z) = \sum_{j=0}^{N-1} a_j z^{-j}$.

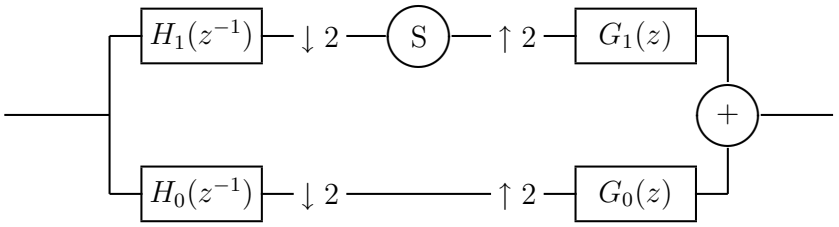


Figure 3.2: Filter bank for wavelet shrinkage on the finest scale.

Discrete Wavelet Shrinkage

Let us review the three steps of wavelet shrinkage in the discrete setting: Figure 3.2 shows the corresponding filter bank for wavelet shrinkage on the finest scale.

1. **Analysis:** In the analysis step, the initial signal is transferred to a wavelet coefficient representation. This decomposition is done with the help of the analysis filters h_0 and h_1 which can be obtained as scaling coefficients of the corresponding scaling function. The filter h_0 plays the role of a low-pass filter, and h_1 plays the role of the corresponding high-pass filter in this context. In addition, both channels are sampled down by leaving out all components with odd index. This is indicated in the filter bank with the symbol $\downarrow 2$.

2. **Shrinkage:** The wavelet coefficients of the signal are shrunken towards zero in this step while the low-frequency components are kept. This is modelled as applying a nonlinear *shrinkage function* $S : \mathbb{R} \to \mathbb{R}$ to each of the wavelet coefficients.

3. **Synthesis:** In this step, the resulting signal is synthesised out of the wavelet coefficients. First an upsampling is used by introducing zeros between each pair of neighbouring signal components. This is written as $\uparrow 2$ here. For the synthesis, the filter pair g_0 and g_1 is used.

We note that the analysis filters h_0 and h_1 are mirrored in our notation. In this figure, also up- and downsampling are included as it is used in classical filter banks. It is well-known [192, 179, 128] that in order to make a perfect reconstruction of the signal possible, the analysis and the synthesis filters have to satisfy the following properties:

$$G_0(z)H_0(z^{-1}) + G_1(z)H_1(z^{-1}) = 2 \quad \text{and} \tag{3.53}$$

$$G_0(z)H_0(-z^{-1}) + G_1(z)H_1(-z^{-1}) = 0 \ . \tag{3.54}$$

For filters with a finite length, one can further show (see [192, p. 120] or [128, Theorem 7.9], for example) that there are numbers $\alpha \neq 0$ and $k \in \mathbb{Z}$ such that

$$G_0(z) = \frac{2}{\alpha} z^{2k+1} H_1(-z^{-1}) \quad \text{and} \quad G_1(z) = -\frac{2}{\alpha} z^{2k+1} H_0(-z^{-1}) \ . \tag{3.55}$$

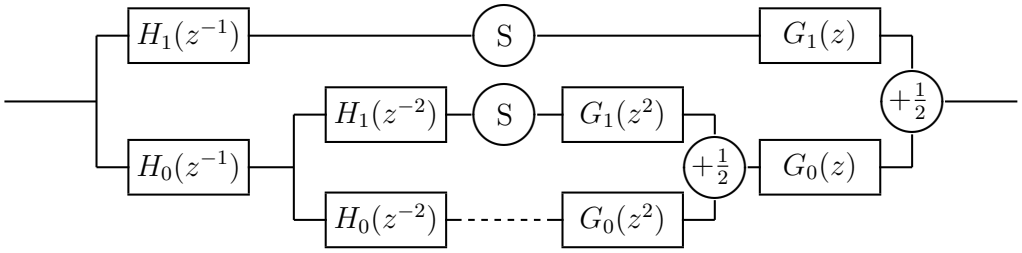


Figure 3.3: Filter bank for translational invariant wavelet shrinkage with multiple scales using the algorithme à trous.

For simplicity, we assume without loss of generality that $\alpha = 2$ and $k = 0$. This gives us the simple relations between analysis and synthesis filters:

$$G_0(z) \;=\; zH_1(-z^{-1}) \quad \text{and} \quad G_1(z) \;=\; -zH_0(-z^{-1}) \;. \tag{3.56}$$

It immediately follows that

$$H_0(z) \;=\; zG_1(-z^{-1}) \;. \tag{3.57}$$

These equations hold for the general biorthogonal case with filters of finite length. In order to have orthonormal filters, we have the further requirement that

$$G_i(z) \;=\; H_i(z) \quad \text{for } i \in \{0, 1\} \tag{3.58}$$

which allows us to determine all four filters with one prototype.

In addition to the filter bank shown in Figure 3.2 we want to achieve translational invariance. This can be done by the algorithme à trous by Holschneider et al. [99, 128] for which the corresponding filter bank is displayed in Figure 3.3. Here, the up- and downsampling is left out, and for the synthesis, the result has to be multiplied with 1/2 at each scale. We see that the analysis and synthesis filters are made wider by inserting zeros in this algorithm. The algorithm as shown in Figure 3.3 will be the basis for our considerations later on.

Let us first take a look at the discretisations of nonlinear diffusion which we will need in this section.

Discrete Higher Order Nonlinear Diffusion

Here we use a discretisation of the nonlinear diffusion equation (2.36) as it has been described in Example 2.26.

The domain Ω will be either the real axis or an interval $\Omega = [0, N-1]$ for $N \in \mathbb{N}$ here. In the case of an interval, we use periodic boundary conditions $u(x+N) = u(x)$ for all $x \in \mathbb{R}$. To discretise the equations, we regard $u \in \mathbb{R}^N$ as a discrete sample on an equidistant grid Ω_h with N pixels and spatial step size 1. As usual, the index set of the pixels is given by $J = \{0, \ldots, N-1\}$.

To approximate the spatial derivatives in (2.36), we use a simple forward difference as finite difference approximation of the first derivative which can be expressed in matrix form as

$$D := \begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -1 & 1 \\ 1 & 0 & \dots & 0 & -1 \end{pmatrix} \tag{3.59}$$

We have already seen this matrix in (2.58) for $h > 0$ in the previous chapter. Its z-transform is $D(z) = z^{-1} - 1$. The transposed matrix D^T then has the z-transform $D^T(z) = z - 1$ and yields an approximation of the negated first derivative with a backward difference.

For time discretisation we use a simple Euler forward scheme. The discrete iterative scheme thus can be written as

$$\begin{aligned} u^0 &= f \\ u^{k+1} &= u^k - \tau (D^T)^p \Phi_{D^p}(u^k) D^p u^k, \qquad k \in \mathbb{N} \ . \end{aligned} \tag{3.60}$$

The diagonal matrix $\Phi_{D^p}(u^k) := \operatorname{diag}\Big(g(|(D^p u^k)_j|)\Big)_{j=0,\dots,N-1}$ stands for the multiplication with the nonlinear diffusivity function and follows the notation in the last chapter.

3.2.2 Discrete Wavelets and Convolution Kernels

This subsection formulates the discrete counterparts to the ideas presented in Subsection 3.1.2. We make the assumption that the wavelet has p vanishing moments io relate the wavelet transform to an approximation of the p-th derivative. In the discrete setting, this condition reads as:

Definition 3.9 (Vanishing Moments)
A signal $f \in \ell^2(\mathbb{Z})$ is said to have $p \in \mathbb{N}$ vanishing moments iff

$$\sum_{n=-\infty}^{\infty} n^j f_n = 0 \qquad \text{for } j \in \{0, \dots, p-1\} \text{ and } \sum_{n=-\infty}^{\infty} n^p f_n \neq 0 \ . \tag{3.61}$$

In the following, we will factorise the z-transform of a wavelet with p vanishing moments such that we obtain a derivative approximation filter and a convolution or smoothing kernel. Since the number of vanishing moments is directly connected with regularity properties, such factorisations are often used in the design of wavelets (see [52, 179, 128, 120], for example). It should also be noticed that the number of vanishing moments of the filter coefficients is the same as the number of (continuous) vanishing moments of the continuous wavelet function, see [128, Theorem 7.4].

Proposition 3.10 (Wavelet Filter Factorisation)
Let $f \in \ell^2(\mathbb{Z})$ be a filter with finite length and p vanishing moments. Then its z-transform can be decomposed as

$$F(z) \;=\; (z-1)^p\, K(z) \;\;,$$

where K is the z-transform of the corresponding filter k which will be understood as smoothing kernel.

Proof: Since f has finite length, the Fourier transform $\hat{f} \in \mathscr{C}^\infty$ is infinitely many times differentiable. The j-th derivative of $\hat{f}$ at the point 0 is then

$$\hat{f}^{(j)}(0) \;=\; (-i)^j \sum_{n=-\infty}^{\infty} n^j\, f_n \tag{3.62}$$

which is the j-th moment of f times the nonzero constant $(-i)^j$. Our assumption about f then reads as $\hat{f}^{(j)} = 0$ for $j \in \{0, \ldots, p-1\}$. This means the Fourier transform of f is a trigonometric polynomial which has a zero of order p in 0. Thus it can be factorised as

$$\hat{f}(\xi) \;=\; (\exp(i\xi) - 1)^p\, K(\exp(i\xi)) \tag{3.63}$$

with a suitable (Laurent-) polynomial K. Replacing $\exp(i\xi)$ by z directly yields the desired factorisation $F(z) = (z-1)^p K(z)$ of the z-transform. □

With the help of this proposition, we can understand the convolution with a wavelet as derivative approximation of a presmoothed signal. We remember that $z-1$ is the z-transform of the finite difference matrix D^T approximating the negated first derivative. Thus $(z-1)^p$ can be used as approximation of $(-1)^p$ times the p-th derivative. This reasoning of understanding the wavelet as derivative of a smoothing kernel is in accordance with the approach in the last section and in [62]. For details on such factorisations, see [128, Section 7.2] for orthogonal wavelets and [128, Section 7.4.2] for the biorthogonal case, for example. Let p and q be the number of vanishing moments of our analysis and synthesis highpass filters H_1 and G_1. Then Proposition 3.10 allows us to write the filters as

$$H_1(z) \;=\; (z-1)^p K^H(z) \quad \text{and} \quad G_1(z) \;=\; (z-1)^q K^G(z) \tag{3.64}$$

where K^H and K^G are the z-transforms of two smoothing kernels k^H and k^G of the synthesis and analysis wavelet. For orthogonal wavelets, we simply have $K^H(z) = K^G(z)$ and $p = q$. With the two relations (3.56) and (3.57) between low- and highpass we see that for the lowpass filters H_0 and G_0, the following relations hold:

$$H_0(z) \;=\; (-1)^q\, z(z^{-1}+1)^q\, K^G(-z^{-1}) \tag{3.65}$$

$$G_0(z) \;=\; (-1)^p\, z(z^{-1}+1)^p\, K^H(-z^{-1}) \tag{3.66}$$

To make these formulae a bit more intuitive, let us now give some examples of how the kernels K^H and K^G look like for commonly used wavelets on the finest scale:

Example 3.11 (Discrete Wavelets and Convolution Kernels)
Haar Wavelet: For the discrete Haar wavelet, we have $H_1(z) = \frac{1}{\sqrt{2}}(z-1)$. The kernel on the finest scale is in this case just a scalar factor $K^H(z) = \frac{1}{\sqrt{2}}$.
Daubechies Wavelets: The Daubechies wavelet [52] with $p = 2$ is represented by the filter

$$H_1(z) \;=\; \frac{1}{4\sqrt{2}}\left(\sqrt{3}-1+(3-\sqrt{3})z-(3+\sqrt{3})z^2+(1+\sqrt{3})z^3\right) \tag{3.67}$$

which can be factorised as $H_1(z) = (z-1)^2 K^H(z)$ leading to

$$K^H(z) \;=\; \frac{1}{4\sqrt{2}}\left(\sqrt{3}-1+(\sqrt{3}+1)z\right) \;. \tag{3.68}$$

Let us shortly say a few words about the differences between this idea and previously sketched approaches for the relation between shrinkage on the finest scale and nonlinear diffusion:

Remark 3.12 (Differences to Previous Approaches)
In contrast to this idea, Weickert et al. [205] have directly considered the wavelet filter H_1 as stencil for a derivative approximation. With a Taylor expansion, one can directly prove that any filter with p vanishing moments yields an approximation of the p-th derivative up to a constant factor. This works well as long as one only considers shrinkage on the finest scale, but it does not help to explain what happens on coarser scales.

Here, we try to model that by separating the derivative approximation from the smoothing kernel which yields a coarse scale approximation of our signal. The complete analogy with the continuous case in the previous section ends at the point where we take a look at the scaled versions of the smoothing kernel. In the continuous setting, the smoothing kernel is a function for that the scaling operation is invertible without loss of information.

In contrast to this, discrete wavelets on coarser scales can change their appearance due to discrete effects. Following [192, Section 3.3], we define:

Definition 3.13 (Wavelets on Larger Scales)
Starting from the filters G_0 and G_1 on the finest scale, we define the wavelet filters $G_0^{(\sigma)}$ and $G_1^{(\sigma)}$ on coarser scales $\sigma \in \mathbb{N}$ as

$$G_0^{(\sigma)}(z) \;=\; \prod_{r=0}^{\sigma-1} G_0(z^{(2^r)}) \quad \textit{and} \quad G_1^{(\sigma)}(z) \;=\; G_1(z^{(2^{\sigma-1})}) G_0^{(\sigma-1)}(z) \;. \tag{3.69}$$

Analogously, we use the same formulae for H_0 and H_1.

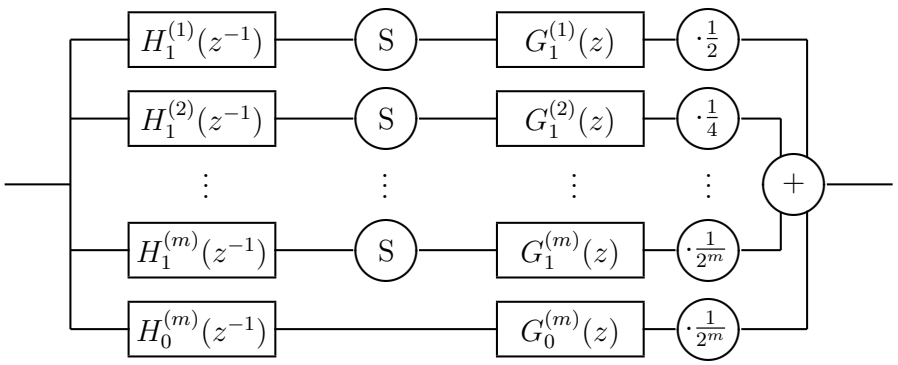


Figure 3.4: Filter bank for translation invariant wavelet shrinkage, written with multiple channels.

The exponents 2^r come from the fact that the algorithme à trous inserts the corresponding number of zeros between two samples of the filter at scale r. In addition, we have to multiply the z-transforms of all filters lying on the path from the input to the middle of the filter bank for H in Figure 3.3, or from the middle to the output for G.

Having these formulae at hand we can rewrite the filter bank in Figure 3.3 with $m+1$ different paths as shown in Figure 3.4. Now we are interested in how the shape of the convolution kernels corresponding to the wavelets in this filter bank changes when the scale increases. Our starting point are the relations (3.69), and we firstly consider the scaling coefficients using the factorisation (3.66):

$$\begin{aligned} G_0^{(\sigma)}(z) &= \prod_{r=0}^{\sigma-1} G_0(z^{(2^r)}) \\ &= \prod_{r=0}^{\sigma-1} \left((-1)^p z^{(2^r)} (z^{-(2^r)}+1)^p \, K^H(-z^{-(2^r)})\right) \\ &= (-1)^{\sigma p} \, z^{(2^\sigma)-1} \left(\sum_{r=0}^{(2^\sigma)-1} z^{-r}\right)^p \prod_{r=0}^{\sigma-1} K^H(-z^{-(2^r)}) \ . \end{aligned} \tag{3.70}$$

We see that the scaling filter on larger scales can be decomposed into four parts: The sign is given by $(-1)^{\sigma p}$, and there is a pure shift $z^{(2^\sigma)-1}$. These two parts do not change the shape of the convolution kernel. This shape is determined by the rightmost two factors: The second one is a product of the kernels k^H with alternating signs and with inserted zeros. This is actually the wavelet-dependent part. The first factor is independent of the wavelet: It is the p times convolution of a box filter of width 2^σ with itself. This can be understood as a B-spline kernel.

Let us see how this decomposition can be made for the wavelet coefficients:

$$\begin{aligned} G_1^{(\sigma)}(z) &= G_1(z^{(2^{\sigma-1})})G_0^{(\sigma-1)}(z) \\ &= (z^{(2^{\sigma-1})}-1)^q\, K^G(z^{(2^{\sigma-1})})\, G_0^{(\sigma-1)}(z) \\ &= (z-1)^q \left(\sum_{r=0}^{2^{\sigma-1}-1} z^r\right)^q K^G(z^{(2^{\sigma-1})})\, G_0^{(\sigma-1)}(z) \\ &= (z-1)^q\,(-1)^{(\sigma-1)p} \left(z^{-(2^{\sigma-1})+1}\right)^{p-1} \left(\sum_{r=0}^{2^{\sigma-1}-1} z^r\right)^{p+q} \cdot \\ &\quad \cdot K^G(z^{(2^{\sigma-1})}) \prod_{r=0}^{\sigma-2} K^H(-z^{-2^r}) \ . \end{aligned}$$

Here, we also analyse the ingredients of this product: The first factor $(z-1)^q$ tells us that the wavelet can be understood as approximation of the q-th derivative (with sign $(-1)^q$). It is the z-transform of the finite difference matrix $(D^T)^q$ defined above. Again, the sign and the shift do not change the shape of the convolution kernel. As for the scaling function, we also find a spline kernel and a wavelet-dependent part here. The spline kernel has the order $p+q$.

Let us now give some examples of commonly used wavelets to see how the related convolution kernels look like:

Example 3.14 (Haar Wavelet on Larger Scales)
We have already seen that for a Haar wavelet, the kernels $K^G(z) = K^H(z) = \frac{1}{\sqrt{2}}$ are just constants. Thus the wavelet on scale σ can be seen as

$$G_1^{(\sigma)}(z) = (-1)^{(\sigma-1)}\,(z-1)\frac{1}{2^{\frac{\sigma}{2}}}\left(\sum_{r=0}^{2^{\sigma-1}-1} z^r\right)^2 \ . \tag{3.71}$$

This means that in complete analogy to the continuous case, the discrete Haar wavelet is the derivative approximation of a hat function. This hat is created by multiplying a box filter with itself. An example for the scale 8 is shown in Figure 3.5.

Example 3.15 (Daubechies Wavelets on Larger Scales)
For some representatives of the family of Daubechies wavelets [52], we display the corresponding kernels obtained by numerical calculations in Figure 3.6. One can see that the smoothing kernels have a shape similar to a Gaussian kernel with a perturbation at the right side where they even change the sign. Daubechies has proven that the Haar wavelets are the only symmetric or antisymmetric orthonormal wavelets with compact support [52], and so it is clear that the corresponding kernels of Daubechies wavelets of higher order can not be symmetric.

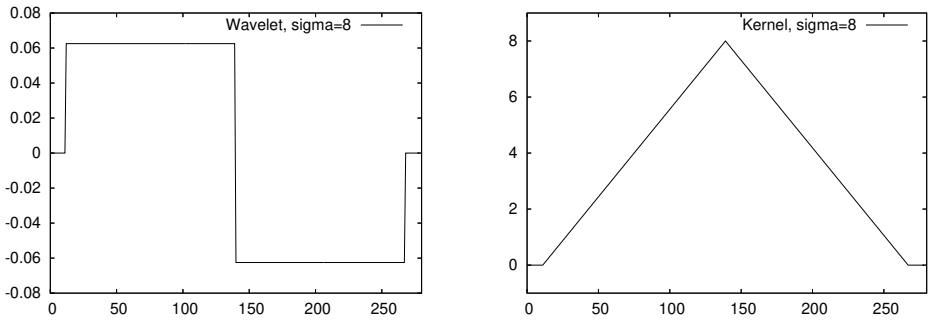


Figure 3.5: Convolution kernel corresponding to the Haar wavelet. *Left column:* Haar wavelet on scale 8. *Right column:* Corresponding smoothing kernel: a hat function.

The following two examples consider the convolution kernels corresponding to biorthogonal filter pairs. For these, it is possible to be symmetric or antisymmetric with compact support, and so the convolution kernels can be symmetric.

Example 3.16 (Compactly Supported Spline Wavelets)
This is our first example of a biorthogonal filter pair. Details on these filters can be found in [128, p. 271], for example. They are displyed in Figure 3.7 have 3 and 7 vanishing moments. We see that the corresponding kernel to h has negative pieces while the kernel derived from $\tilde{h}$ is positive and has the shape of a Gaussian.

Example 3.17 (Perfect Reconstruction Filters of Most Similar Length)
The second example for biorthogonal filters can be found in [128, p. 273], for example. Here we have 4 vanishing moments for both filters that are displayed in Figure 3.8. The filter corresponding to $\tilde{h}$ has some small negative parts.

Regardless the shape of the convolution kernels, it will be important for our considerations in the next subsection that we can write the analysis and the synthesis wavelet as

$$G_1^{(\sigma)}(z) \;=\; (1-z)^q \, K^{G,(\sigma)}(z) \quad \text{and} \quad H_1^{(\sigma)}(z) \;=\; (1-z)^p \, K^{H,(\sigma)}(z) \;. \tag{3.72}$$

We use the notions $K^{G,(\sigma)}$ and $K^{H,(\sigma)}$ to denote the corresponding convolution kernels on scale σ we have derived above. With the finite difference matrices introduced in (3.59), we can rewrite this in matrix notation:

$$G_1^{(\sigma)} \;=\; (D^q)^T \; K^{G,(\sigma)} \quad \text{and} \quad H_1^{(\sigma)} \;=\; (D^p)^T \; K^{H,(\sigma)} \;. \tag{3.73}$$

We are going to use these equations in the next section to rewrite iterated wavelet shrinkage as discretisation of an integrodifferential equation.

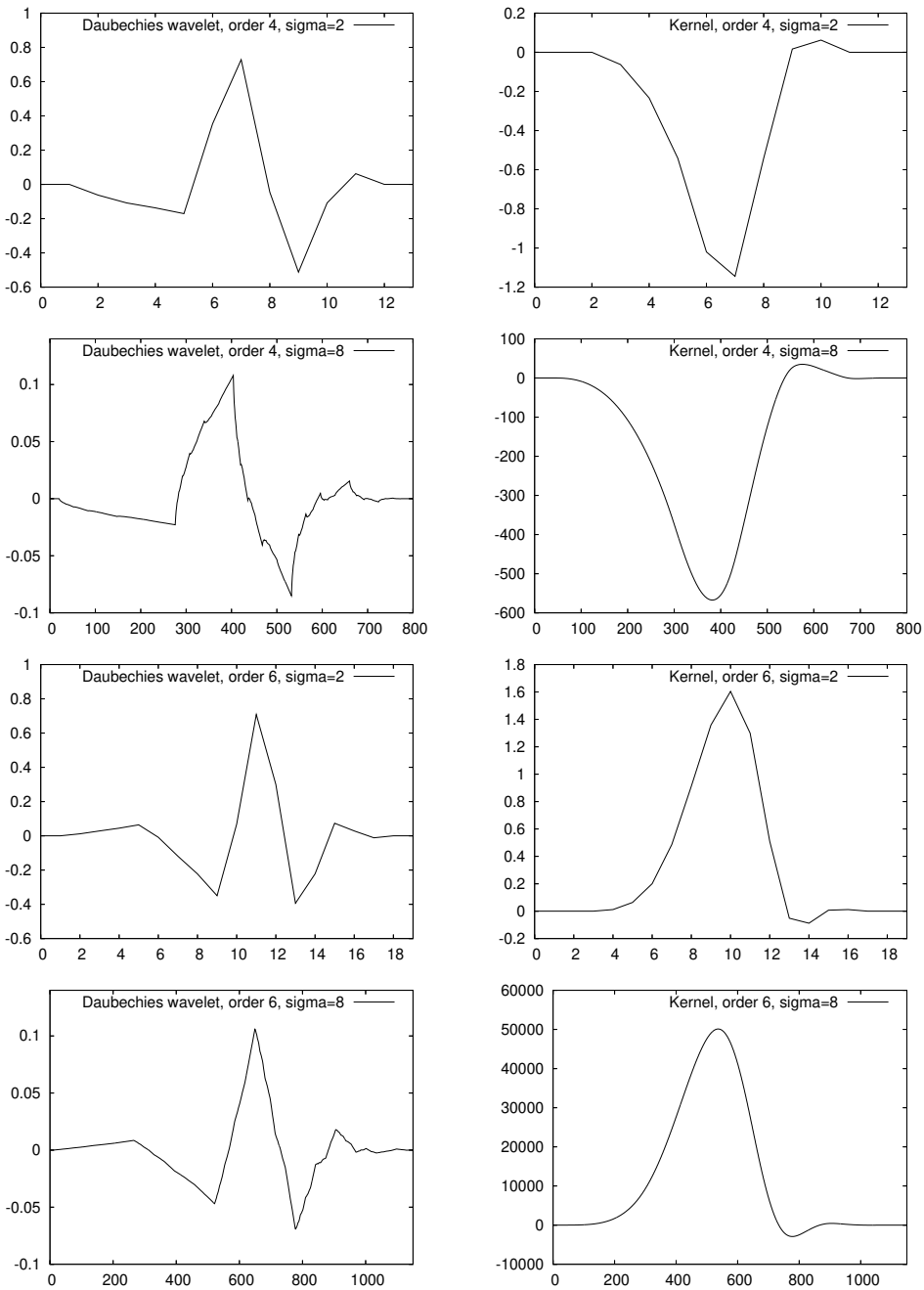


Figure 3.6: Convolution kernels corresponding to Daubechies wavelets on larger scales. *Left column:* Daubechies wavelets of orders 4 and 6 on scales 2 and 8. *Right column:* Corresponding smoothing kernels. The scaling comes from the fact that wavelets are normalised with respect to the ℓ^2-norm.

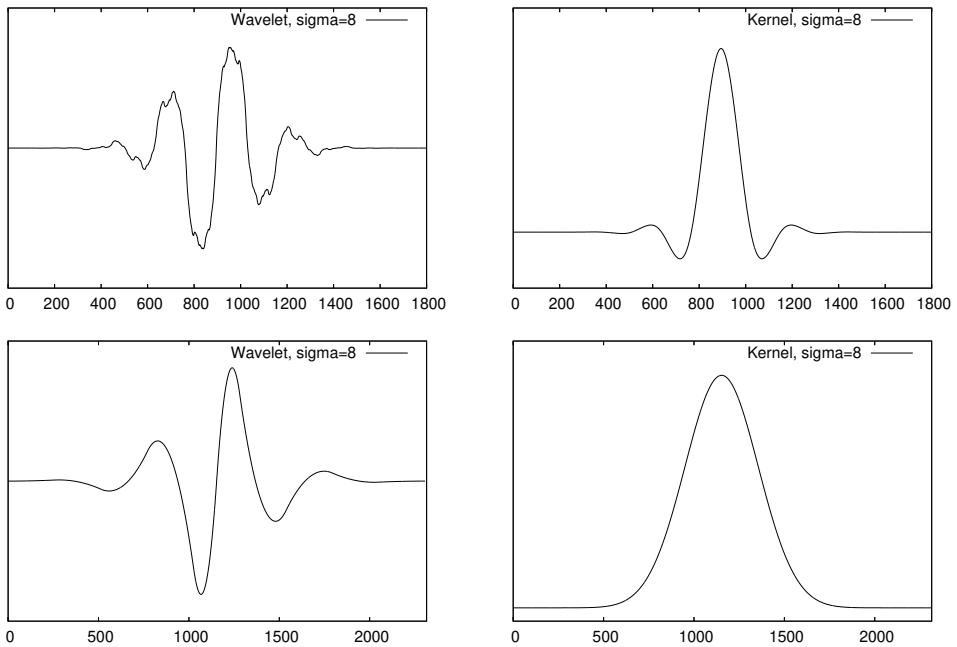


Figure 3.7: Convolution kernels corresponding to compactly supported spline wavelets on scale 8. *Top left:* Filter h with 3 vanishing moments. *Top right:* Filter $\tilde{h}$ with 7 vanishing moments. *Right column:* Corresponding smoothing kernels.

3.2.3 Relations Between Both Methods

As already indicated by Figure 3.4 let us rewrite wavelet shrinkage in the following way:

$$u \;=\; \sum_{\sigma=1}^{m} \frac{1}{2^\sigma}\, G_1^{(\sigma)} S\left(\left(H_1^{(\sigma)}\right)^T f\right) \;+\; \frac{1}{2^m}\, G_0^{(m)} \left(H_0^{(m)}\right)^T f \;. \tag{3.74}$$

In this section, $f, u \in \mathbb{R}^N$ are vectors, and $H_0^{(\sigma)}$ denotes the N-dimensional circulant matrix corresponding to the filter $H_0^{(\sigma)}(z)$. The analysis matrices are transposed to reflect the fact that we have used $H_i(z^{-1})$ for $i = 0, 1$ on the analysis part of our filter banks in Figures 3.2, 3.3, and 3.4. The function S is meant to act componentwise on the vector entries here.

Without shrinking the coefficients, the filter bank will allow for a perfect reconstruction: We have

$$f \;=\; \sum_{\sigma=1}^{m} \frac{1}{2^\sigma}\, G_1^{(\sigma)} \left(H_1^{(\sigma)}\right)^T f \;+\; \frac{1}{2^m}\, G_0^{(m)} \left(H_0^{(m)}\right)^T f \tag{3.75}$$

for all $f \in \mathbb{R}^N$. As already done in the last section and similar to [140, 62] we use (3.19) to rewrite our shrinkage function as $S(x) = (1 - \tau g(|x|))x$. Plugging

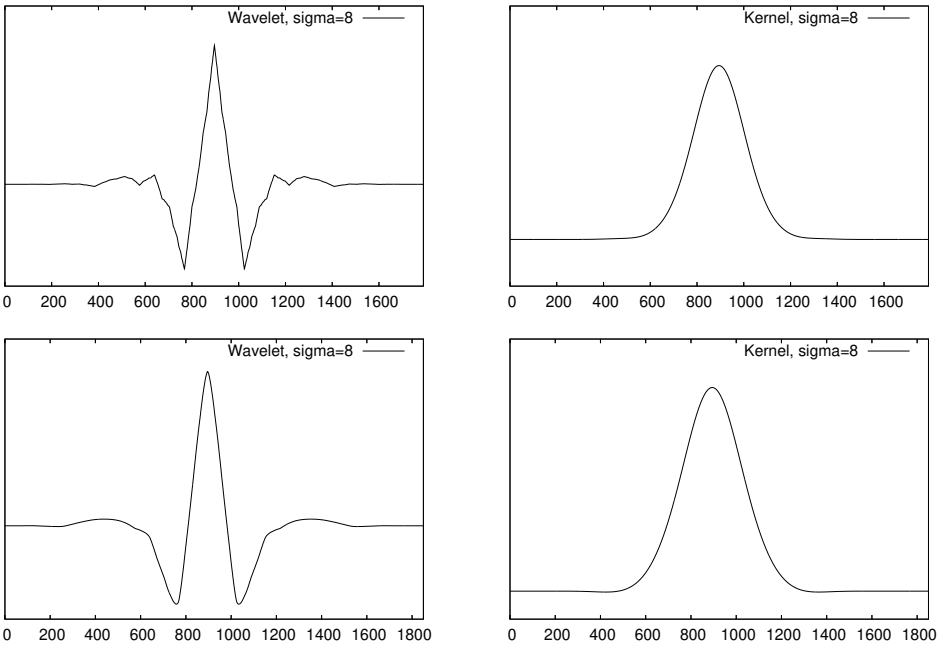


Figure 3.8: Convolution kernels corresponding to perfect reconstruction filters of most similar length on scale 8. *Left column:* Analysis and synthesis filter. *Right column:* Corresponding smoothing kernels.

this into (3.74) we obtain

$$\begin{aligned} u &= \sum_{\sigma=1}^{m} \frac{1}{2^\sigma} G_1^{(\sigma)} \left(H_1^{(\sigma)}\right)^T f + \frac{1}{2^m} G_0^{(m)} \left(H_0^{(m)}\right)^T f \\ &\quad - \tau \sum_{\sigma=1}^{m} \frac{1}{2^\sigma} G_1^{(\sigma)} \Phi\left(\left(H_1^{(\sigma)}\right)^T f\right) \left(H_1^{(\sigma)}\right)^T f \ . \end{aligned} \tag{3.76}$$

The matrix Φ is a diagonal matrix such that $\Phi(v) := \operatorname{diag}\Big(g(|v_j|^2)\Big)_{j\in J}$. Property (3.75) shows that the first part gives a reconstruction of the initial signal f, and we obtain

$$u = f - \tau \sum_{\sigma=1}^{m} \frac{1}{2^\sigma} G_1^{(\sigma)} \Phi\left(\left(H_1^{(\sigma)}\right)^T f\right) \left(H_1^{(\sigma)}\right)^T f \tag{3.77}$$

for one single shrinkage step. Performing several steps of shrinkage gives an iterative scheme

$$\begin{aligned} u^0 &= f \\ u^{k+1} &= u^k - \tau \sum_{\sigma=1}^{m} \frac{1}{2^\sigma} G_1^{(\sigma)} \Phi\left(\left(H_1^{(\sigma)}\right)^T u^k\right) \left(H_1^{(\sigma)}\right)^T u^k \ , \quad k \in \mathbb{N} \end{aligned} \tag{3.78}$$

which has a very similar structure than the discretisation of the nonlinear diffusion equation (3.60). Using (3.73), the iteration rule can be written as

$$u^{k+1} \;=\; u^k - \tau \sum_{\sigma=1}^{m} \frac{1}{2^\sigma} \left(D^q\right)^T K^{G,(\sigma)} \, \Phi\left(D^p \left(K^{H,(\sigma)}\right)^T u^k\right) \, D^p \left(K^{H,(\sigma)}\right)^T u^k \;. \tag{3.79}$$

This equation is the discrete equivalent to (3.23) and describes how iterated discrete wavelet shrinkage can be seen as discretisation of an integrodifferential equation.

Interpretation. As in the continuous case, we see that the two differences between discrete wavelet shrinkage and nonlinear diffusion is that all derivatives are presmoothed and that we sum over all scales σ. In contrast to the continuous considerations in the last section, we have worked with two different kernels here to allow for biorthogonal wavelets. This can lead to equations where the inner and the outer derivative order differ. In the PDE-based image processing context, we have seen similar ideas, but without presmoothing, in the filters of Tumblin and Turk [189] and Wei [195] described in Subsection 1.2.2. These approaches even go one step further: They do not only allow the derivative orders in front of the nonlinear function and behind to be different, but also the argument can be a third one. By the construction of (3.23) and (3.79) this is not included in the framework shown here.

Remark 3.18 (Orthogonal Wavelets)
In the case of orthogonal wavelets, (3.79) simplifies to

$$u^{k+1} \;=\; u^k - \tau \sum_{\sigma=1}^{m} \frac{1}{2^\sigma} \left(D^p\right)^T K^{H,(\sigma)} \, \Phi\left(D^p \left(K^{H,(\sigma)}\right)^T u^k\right) \, D^p \left(K^{H,(\sigma)}\right)^T u^k \tag{3.80}$$

since we have $p = q$, and also the kernels are identical. Besides the smoothing kernels and the sum over all scales, this is identical to the explicit discretisation of a higher order nonlinear diffusion equation (2.50). Since the outer matrices are the adjoints of the inner ones, this approach can be understood as coming from an energy function of the form

$$E(u) \;=\; \sum_{i \in J} (u_i - f_i)^2 \;+\; \alpha \sum_{\sigma=1}^{m} \frac{1}{2^\sigma} \sum_{i \in J} \Psi\Big(\left(D^p K^{H,(\sigma)} u\right)_i^2 \Big) \;. \tag{3.81}$$

This is the discrete analogon to (3.24). For biorthogonal wavelets, this is not possible.

3.2.4 Generalisation to Higher Dimensions

So far, the ideas in this chapter have been considered in one spatial dimension only. Let us now say a few words to the two-dimensional case. For one single

scale of Haar wavelet shrinkage, relations to nonlinear diffusion equations have been discussed by Mrázek and Weickert [137, 138], for example. The description in [138] already indicates that a direct transfer of the ideas presented here can be more technical, but yields very good results for a the example of the Haar wavelet. Here we follow the easier strategy sketched in [137] and apply it not only to one scale of Haar wavelet shrinkage, but to multiple scales with general biorthogonal filters.

It is common practice to use tensor product wavelets for the processing of two-dimensional images, see [128, Subsections 7.7.2 and 7.7.3] or [89, Section 7.5], for example. At the example of one-dimensional analysis scaling coefficients h_0 and wavelet coefficients h_1, the tensor product analysis filters h_s, h_h, h_v, and h_d in 2-D read as

$$\begin{aligned} h_s(i,j) &:= h_0(i)h_0(j) \ , \qquad & h_h(i,j) &:= h_1(i)h_0(j) \ , \\ h_v(i,j) &:= h_0(i)h_1(j) \ , \qquad & h_d(i,j) &:= h_1(i)h_1(j) \ . \end{aligned}$$

Here, the subscript s stands for scaling function, h for the horizontal, v for the vertical, and d for the diagonal wavelet. The same definition applies for the synthesis coefficients with g instead of h. It is a classical result that these filters on multiple scales yield a biorthogonal family in 2-D (see the references given above). In analogy to (3.75), the perfect reconstruction property for m scales in 2-D can be formulated as

$$f \ = \ \sum_{\sigma=1}^{m} \frac{1}{4^\sigma} \Big(\sum_{\delta \in \{h,v,d\}} G_\delta^{(\sigma)} \left(H_\delta^{(\sigma)} \right)^T f \Big) \ + \ \frac{1}{4^m} G_s^{(m)} \left(H_s^{(m)} \right)^T f \tag{3.82}$$

with the help of the corresponding matrices to the filters defined above. One step of shrinkage then reads as

$$u \ = \ \sum_{\sigma=1}^{m} \frac{1}{4^\sigma} \Big(\sum_{\delta \in \{h,v,d\}} G_\delta^{(\sigma)} S^\delta \Big(\left(H_\delta^{(\sigma)} \right)^T f \Big) \Big) \ + \ \frac{1}{4^m} G_s^{(m)} \left(H_s^{(m)} \right)^T f \tag{3.83}$$

with the shrinkage functions S^h, S^v, and S^d applied to the corresponding wavelet coefficients.

To give a motivation for using different shrinkage functions in the three directions, we take a look at the approximation properties of the wavelet coefficients in 2-D. Convolution of an image with the filters given above can also be understood as derivative approximation with presmoothing while the derivative order and the smoothing kernel depend on h_1 and h_0. For example, let p be the number of vanishing moments of h_1. Convolution of a discrete image u with h_h and h_v approximates presmoothed p-th derivatives of u in x- and y-direction. The filter h_d yields the approximation of the derivative $\partial_x^p \partial_y^p u$ with additional smoothing. That means this derivative in diagonal direction has twice the order than the other ones.

This fact suggests to follow a shrinkage rule as described in [137] to improve rotational invariance. Inspired by nonlinear diffusion filtering, it is suggested there to couple the horizontal and vertical coefficients in the argument of the shrinkage function and not to shrink the diagonal ones at all. Let w_h, w_v, and w_d stand for the wavelet coefficients in horizontal, vertical and diagonal direction at a given scale and position. The corresponding shrinkage functions applied to the horizontal, vertical and diagonal coefficients can be written down as:

$$S^h(w_h, w_v) := w_h(1 - \tau g(w_h^2 + w_v^2)) \ , \tag{3.84}$$

$$S^v(w_h, w_v) := w_d(1 - \tau g(w_h^2 + w_v^2)) \ , \tag{3.85}$$

$$S^d(w_d) := w_d \ . \tag{3.86}$$

In opposition to [137] we avoid the additional factor 4 in front of the function g here. This factor can be explained as compensation of the factor $\frac{1}{4}$ appearing in (3.82) and (3.83) together with the fact that only the finest scale is considered in [137]. We avoid using the factor here since we work on multiple scales and would need different factors for each scale. We prefer to use the same shrinkage function on all scales instead.

With these shrinkage functions and the perfect reconstruction property (3.82), wavelet shrinkage (3.83) can be transformed into

$$u = f - \tau \sum_{\sigma=1}^{m} \frac{1}{4^\sigma} \left(G_h^{(\sigma)} \, \Phi_h^{(\sigma)} \left(H_h^{(\sigma)} \right)^T f + G_v^{(\sigma)} \, \Phi_v^{(\sigma)} \left(H_v^{(\sigma)} \right)^T f \right) \ . \tag{3.87}$$

Here, $\Phi_h^{(\sigma)}$ and $\Phi_v^{(\sigma)}$ represent the pointwise multiplication of the wavelet coefficients in horizontal and vertical direction on scale σ with diffusivity g. As defined in (3.84), this diffusivity depends on the squared sum of the horizontal and vertical wavelet coefficients at the corresponding position and scale. Understood as discretisation of an integrodifferential equation, one would use it iteratively yielding

$$u^{k+1} = u^k - \tau \sum_{\sigma=1}^{m} \frac{1}{4^\sigma} \left(G_h^{(\sigma)} \, \Phi_h^{(\sigma)} \left(H_h^{(\sigma)} \right)^T u^k + G_v^{(\sigma)} \, \Phi_v^{(\sigma)} \left(H_v^{(\sigma)} \right)^T u^k \right) \ . \tag{3.88}$$

This is the 2-D analogue of (3.79).

Example 3.19 (Orthogonal Wavelets in 2-D)
Let us consider the case of orthogonal wavelets, i. e. $G_h = H_h$ and $G_v = H_v$, with p vanishing moments. If we neglect the presmoothing introduced by the wavelets, such a shrinkage process is connected via the reasoning given above to a continuous equation of the form

$$\begin{aligned} \partial_t u = (-1)^{p+1} \Big(& \partial_x^p \left(g(|\partial_x^p u|^2 + |\partial_y^p u|^2) \, \partial_x^p u \right) \\ & + \partial_y^p \left(g(|\partial_x^p u|^2 + |\partial_y^p u|^2) \, \partial_y^p u \right) \Big) \end{aligned} \tag{3.89}$$

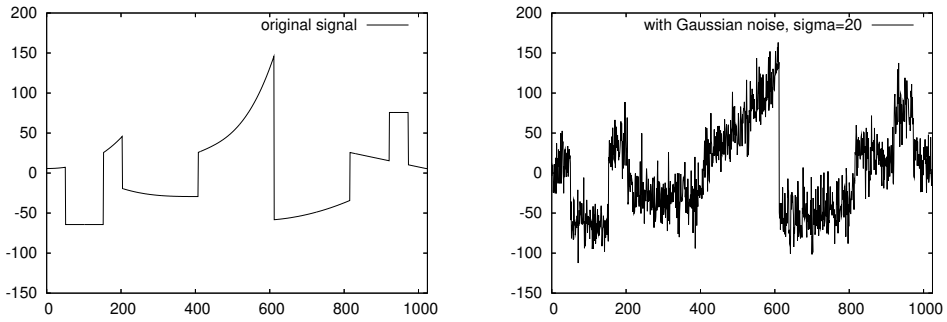


Figure 3.9: Test signals. *Left:* Piecewise polynomials signal with 1024 pixels. *Right:* With additive Gaussian noise, standard deviation 20.

which only considers the derivatives with respect to the coordinate axes. For $p = 1$, this is the classical Perona-Malik equation. Nevertheless, for higher derivative orders $p > 1$, it does not correspond to the higher order nonlinear diffusion equations (2.35) described in Chapter 2: The mixed derivatives of order p are not represented by the wavelets.

3.2.5 Numerical Experiments

In this subsection we want to investigate experimentally the differences between classical nonlinear diffusion filtering and the integrodifferential equations related to wavelet shrinkage described in this chapter. In 1-D, we perform detailed qualitative comparisons for the denoising of a signal with additive Gaussian noise. Experiments for image simplification in 2-D show that the same effects appear in this case, too.

Let us first describe our experiments in 1-D: Figure 3.9 shows the test signal we are going to use and a noisy version of it with Gaussian noise of standard deviation 20. The test signal has been published as `piecepoly` in the `Wavelab` software package [1]. The implementations used here are not based on `Wavelab`, but have been written in `C`.

In our first experiment, we perform a qualitative comparison for denoising between diffusion and presmoothed versions. These correspond to the equation

$$u^{k+1} \;=\; u^k - \tau \, (D^p)^T \, K^H \, \Phi \left(D^p \left(K^H \right)^T u^k \right) \, D^p \left(K^H \right)^T u^k \tag{3.90}$$

where we have set the order $p = 1$ and use a hat function as kernel in the matrices K^H. As we have seen in Subsection 3.2.2, this corresponds to Haar wavelets. The kernel length is chosen as $l = 2\sigma$ here. We have used one single scale for presmoothing, and thus in contrast to (3.80), there is no sum and no weight factor

[1] Wavelab is available under the address http://www-stat.stanford.edu/~wavelab/.

Scale σ	Error in ℓ^1-norm			Error in ℓ^2-norm		
	error	λ	iterations	error	λ	iterations
1	**2806.16**	**1.02**	**4593**	**143.91**	**1.67**	**1265**
2	5208.78	0.10	247000	232.74	0.10	233000
3	6671.34	0.10	351000	292.34	0.10	263000

Table 3.2: Error norms for denoising results with presmoothed diffusion and one single scale.

Largest scale	Error in ℓ^1-norm			Error in ℓ^2-norm		
$\sigma = 2^k$	error	λ	iterations	error	λ	iterations
$k = 0$	2805.57	1.02	9197	**143.62**	**1.67**	**2604**
$k = 1$	2891.98	1.47	1904	145.59	2.11	677
$k = 2$	**2782.59**	**2.39**	**495**	145.03	3.57	200
$k = 3$	2857.78	4.02	153	146.65	4.95	95
$k = 4$	3072.26	6.36	53	149.52	5.84	61
$k = 5$	3260.64	8.95	27	153.52	6.47	48

Table 3.3: Error norms for denoising results using presmoothed diffusion on dyadic scales.

on the right-hand side. The parameters have been optimised in order to obtain minimal errors in both the ℓ^1- and ℓ^2-norms. The optimal parameters and the corresponding mimimal error measures can be found in Table 3.2. We see that the minimal errors are obtained for classical nonlinear diffusion filters without presmoothing. To visualise the differences and find an explanation for this effect, the corresponding signals are displayed in Figure 3.10. It is clearly visible that using one signal presmoothing kernel for all derivatives leads to artefacts: The process is not able to remove the noise on the small scales which leads to oscillations. Only the general shape of the signal is restored for larger scales. This is in accordance with the results reported by Scherzer and Weickert [165].

In our second experiment, we do not only filter with one larger scale, but involve all dyadic scales $\sigma = 2^l$ for $l = 0, \dots, k$ and use (3.80) for filtering. The corresponding optimal error measures are shown in Table 3.3. We have used a time step size $\tau = 1/2$ for all experiments in 1-D. In this case we see that using larger scales does not influence the minimal error as strong as for one single scale. For the ℓ^1-error, it is even possible to obtain better values by using $k = 2$ in this example. We notice that using only the finest scale requires half the number of iterations than in the first experiment: This comes from the additional factor $\frac{1}{2}$ in (3.80) on the finest scale which was not present in the last experiment. The necessary number of iterations reduces by two orders of magnitude by involving larger scales. This can be understood as approximative numerical methods for speeding up the process. The corresponding signals are shown in Figure 3.11. We

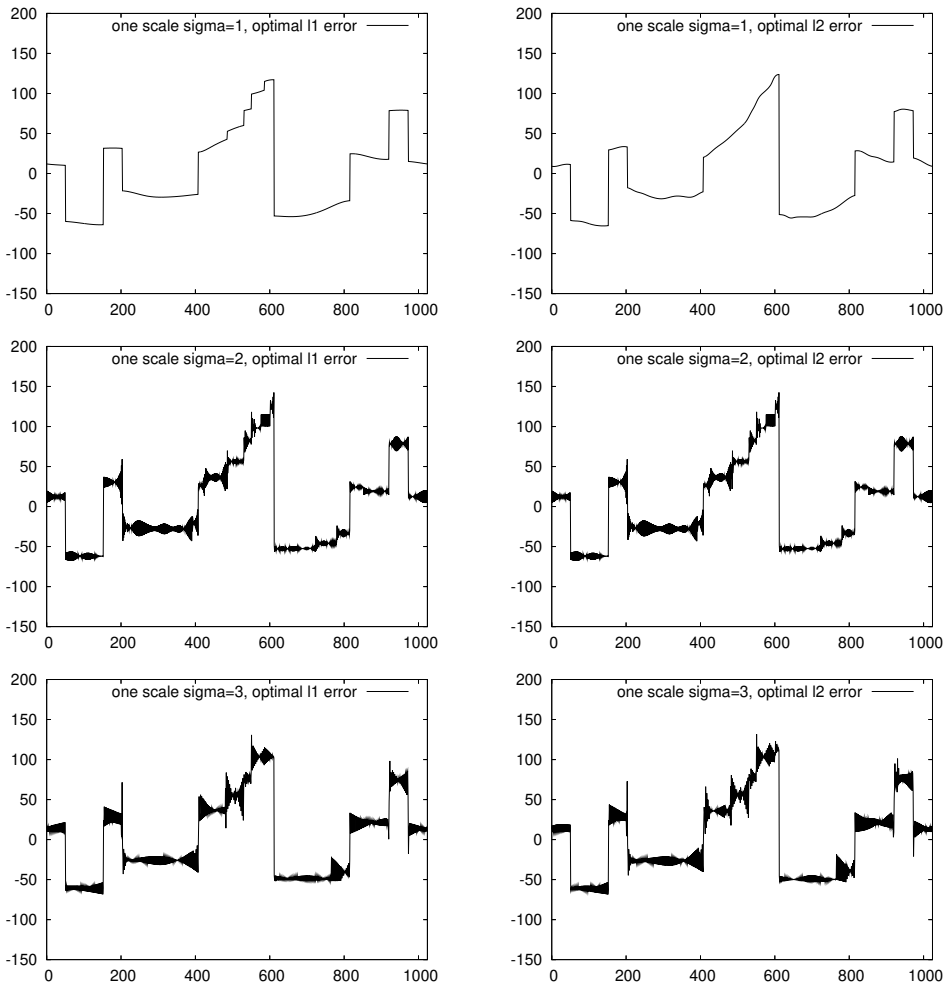


Figure 3.10: Denoising results with presmoothed diffusion and one single scale σ. *Left column:* Results with optimal ℓ^1-error. *Right column:* Results with optimal ℓ^2-error. *Top row:* $\sigma = 1$. *Middle row:* $\sigma = 2$. *Bottom row:* $\sigma = 3$.

see that for larger scales, some smaller artefacts appear. Nevertheless, it seems that the presence of smaller scales in the right-hand side can help to suppress most of them.

In our 2-D experiment, we also display results for smoothing on one larger scale and on all dyadic scales. With one larger scale, we use the filter

$$u^{k+1} \;=\; u^k - \tau \left(K_h^{(\sigma)} \, \Phi_h^{(\sigma)} \left(K_h^{(\sigma)} \right)^T u^k + K_v^{(\sigma)} \, \Phi_v^{(\sigma)} \left(K_v^{(\sigma)} \right)^T u^k \right) \quad . \tag{3.91}$$

This corresponds to (3.88) where the factor and the sum on the right-hand side are left out. We use $p = 1$ and hat functions in the directions of the derivative and box filters in the other direction which implements tensor product Haar wavelets. Figure 3.12 shows the resulting images if we fix all parameters and only vary the scale. We see that using larger scales only introduces artefacts in the image which can be compared to those appearing in the 1-D case also.

For involving all scales we directly use (3.88). Some results for involving all dyadic scales up to a certain order are displayed in Figure 3.13. Here we see that more and more small details are removed by using the larger scales while the artefacts are suppressed.

3.3 Summary

The goal of this chapter was to investigate the relation between continuous multiscale wavelet shrinkage on the one hand and nonlinear diffusion filters of arbitrary order and their variational counterparts on the other hand. This has been achieved by deriving novel integrodifferential equations from multiscale wavelet shrinkage. To this end we exploited the fact that wavelets with a finite number of vanishing moments represent smoothed derivative operators. The resulting integrodifferential equations differ from their nonlinear diffusion counterparts by the additional presmoothing of derivatives and integration over a continuum of scales. Moreover, they can be rewritten as a weighted average of pseudodifferential equations. In the discrete setting, we have studied the shape of the corresponding convolution kernels for larger scales. We have extended the considerations to biorthogonal wavelets: Here, the corresponding integrodifferential equations are no longer related to diffusion equations, but to more general PDE models like the methods by Tumblin and Turk [189]. Using tensor product wavelets and special shrinkage rules for rotational invariance, the relations have been carried over to the 2-D case. Numerical experiments have shown that presmoothed nonlinear diffusion on one single larger scale gives worse results than classical nonlinear diffusion. On the other hand, involving all dyadic scales up to a certain order, as it is done in wavelet shrinkage, almost keeps the good quality and significantly reduces the number of iterations needed. In that sense, it can be understood as numerical method for multiscale approximation of a nonlinear diffusion equation.

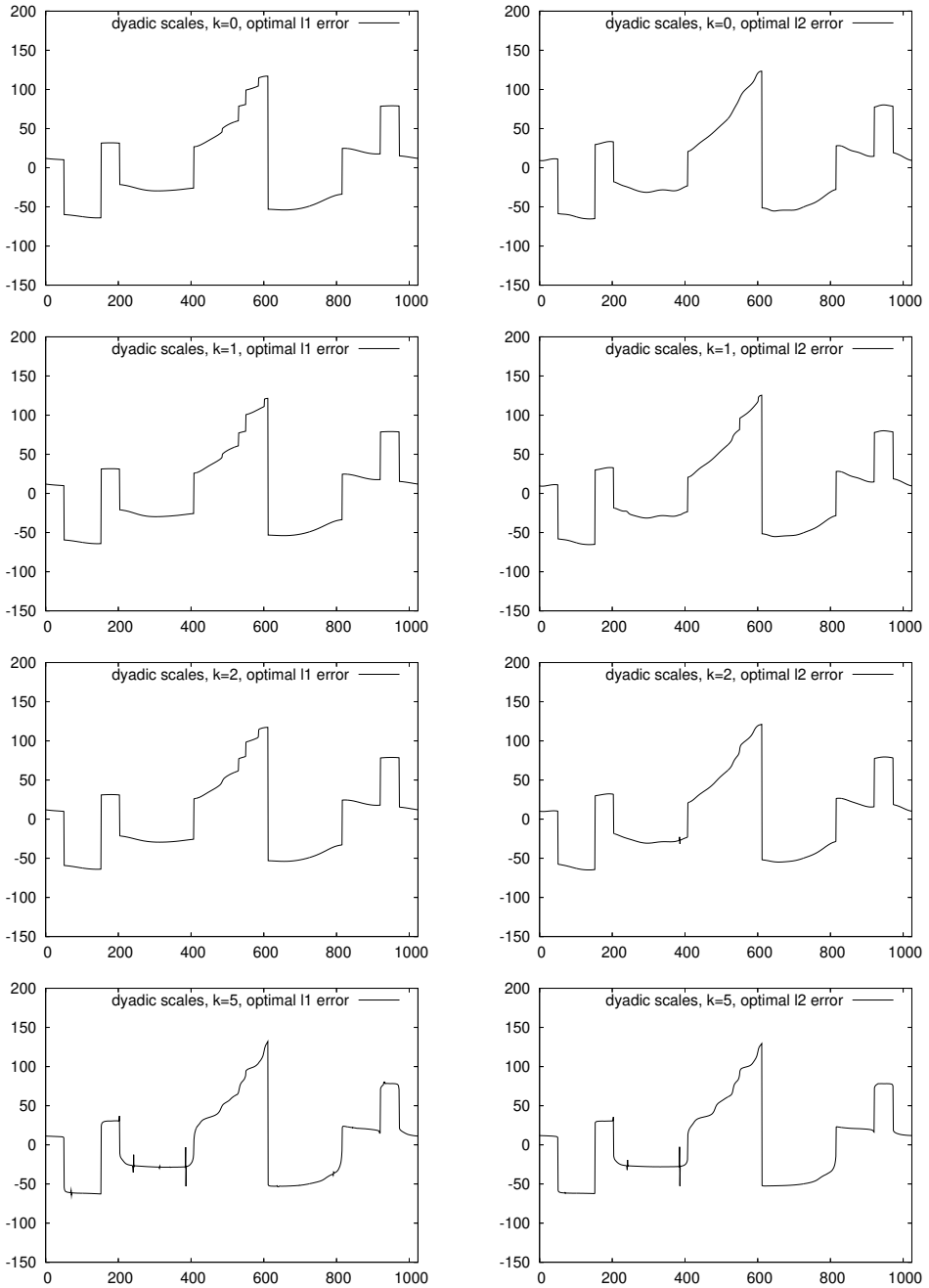


Figure 3.11: Denoising results with presmoothed diffusion and dyadic scale up to $\sigma = 2^k$. *Left column:* Results with optimal ℓ^1-error. *Right column:* Results with optimal ℓ^2-error. *Top row:* $k = 0$. *Second row:* $k = 1$. *Third row:* $k = 2$. *Bottom row:* $k = 5$.

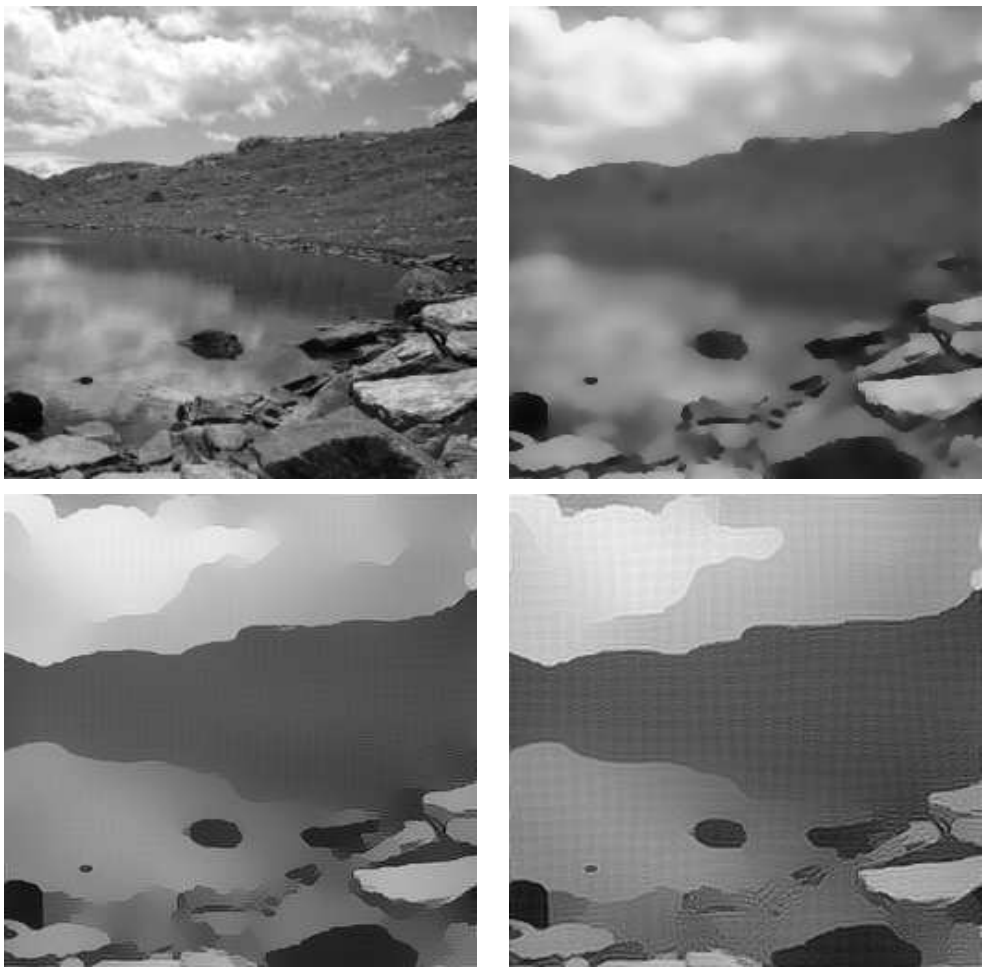


Figure 3.12: Image simplification results with presmoothed diffusion and one single scale σ, Perona-Malik diffusivity $g(s^2) = 1/(1 + s^2/\lambda^2)$ with $\lambda = 10$, and stopping time $t = 10$. *Top left:* Original image, 256×256 pixels. *Top right:* $\sigma = 1$. *Bottom left:* $\sigma = 2$. *Bottom right:* $\sigma = 3$.

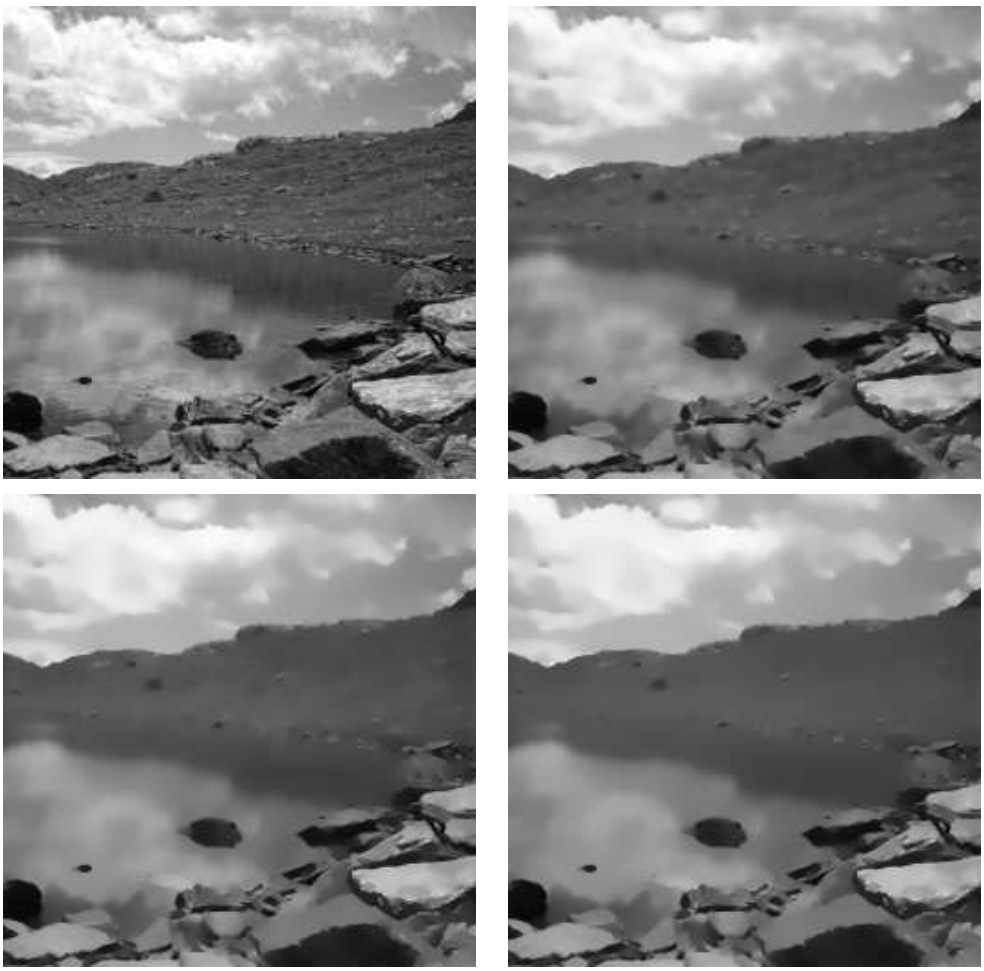


Figure 3.13: Image simplification results with presmoothed diffusion, $g(s^2) = (1 + s^2/\lambda^2)^{-1}$ for $\lambda = 10$, stopping time $t = 10$, and dyadic scales up to $\sigma = 2^k$. *Top left:* Original image, 256×256 pixels. *Top right:* $k = 0$. *Bottom left:* $k = 1$. *Bottom right:* $k = 5$.

Chapter 4

Adaptive Averaging and PDEs

After discussing the connections between wavelet shrinkage and PDE-based filters, we are now interested in weighted averaging schemes and their relation to nonlinear diffusion equations.

We have already sketched some previous work in this direction in Subsection 1.4.1. Especially we would like to mention the ideas of Buades et al. [20, 19]. They have related many neighbourhood filters in a continuous representation to their corresponding PDE methods. In contrast to their approach, we are going to start with discrete neighbourhood filters as they are implemented in practice. This chapter is based on [60, 61].

In Section 4.1 we start with a fully discrete averaging filter and describe how a scaling limit of it can be related to an accelerated variant of the Perona-Malik filter (1.8). These ideas are extended to the two-dimensional case in Section 4.2 in several ways: The averaging filter with very small neighbourhood considered in Subsection 4.2.1 leads to a scaling limit which is not rotationally invariant. Nevertheless, it can be seen as crude approximation of an anisotropic filter similar to the integration model for anisotropic diffusion [197]. The same process can be obtained as scaling limit of an iterated bilateral filter as shown in Section 4.2.2. An alternative way to transfer the 1-D scaling limit to 2-D is presented in Section 4.3 and leads to a family of PDEs which will be called *generalised mean curvature motion* (GMCM) here. This family of filters is part of the very general framework of Carmona and Zhong [30] for filters depending on feature directions. We display practical properties of all presented methods in Section 4.4 with several numerical examples and compare them to classical models. The chapter is concluded with a summary in Section 4.5.

4.1 Averaging Filters and Scaling Limits in 1-D

In this section, we start with a simple discrete averaging filter and derive the corresponding scaling limit. The differences of this PDE model to the Perona-

Malik equation will be described as well as properties of a possible discretisation.

To gain a first intuition, we start with the one-dimensional case. After deriving the scaling limit, we discuss its properties and consider the stability of suitable discretisations.

4.1.1 Derivation of the Scaling Limit

In contrast to the averaging processes considered by Saint-Marc et al. [161] and Barash [11] that we have seen in Subsection 1.4.1, we include the normalisation of the averaging weights in each position explicitly here. Further we specify the arguments of the nonlinear function g in the discrete setting. We will see that due to the normalisation, we will not obtain exactly the Perona-Malik equation as scaling limit later on.

We start with the consideration of an iterative weighted averaging filter of the form

$$\begin{aligned} u_i^0 &= f_i \ , \\ u_i^{k+1} &= \frac{g\left(\left|\frac{u_{i+1}^k - u_i^k}{h}\right|^2\right) u_{i+1}^k + g\left(\left|\frac{u_{i-1}^k - u_i^k}{h}\right|^2\right) u_{i-1}^k}{g\left(\left|\frac{u_{i+1}^k - u_i^k}{h}\right|^2\right) + g\left(\left|\frac{u_{i-1}^k - u_i^k}{h}\right|^2\right)} \end{aligned} \tag{4.1}$$

where $f \in \mathbb{R}^N$ is an initial signal and $u^k \in \mathbb{R}^N$ denotes the processed signal at iteration $k \in \mathbb{N}$. For each pixel u_i^{k+1}, the filter takes the direct neighbours u_{i-1}^k and u_{i+1}^k into account for averaging. At the boundaries, we assume mirroring boundary conditions, that means we introduce two artificial pixels $u_{-1}0^k$ and u_N^k with $u_{-1}^k := u_0^k$ and $u_N^k := u_{N-1}^k$. Typically one chooses a decreasing positive function g such that the denominator can not be zero. This also implies that we always have convex combinations which guarantees a maximum-minimum principle for the filter. One may use e.g. the same function g as the diffusivities in nonlinear diffusion filtering [154], for instance $g(s^2) = (1 + s^2/\lambda^2)^{-1}$. We observe that the weights depend on the grey value distance between the pixel and its direct neighbours divided by the spatial distance $h > 0$ between the two pixels.

We introduce the abbreviations $g_{i+\frac{1}{2}}^k := g\left(\left|\frac{u_{i+1}^k - u_i^k}{h}\right|^2\right)$ and rewrite (4.1) as

$$u_i^{k+1} = \frac{g_{i+\frac{1}{2}}^k u_{i+1}^k + g_{i-\frac{1}{2}}^k u_{i-1}^k}{g_{i+\frac{1}{2}}^k + g_{i-\frac{1}{2}}^k} \tag{4.2}$$

$$= u_i^k + \frac{g_{i+\frac{1}{2}}^k (u_{i+1}^k - u_i^k) - g_{i-\frac{1}{2}} (u_i^k - u_{i-1}^k)}{g_{i+\frac{1}{2}}^k + g_{i-\frac{1}{2}}^k} \tag{4.3}$$

$$= u_i^k + \frac{\frac{1}{h}\left(g_{i+\frac{1}{2}}^k \frac{u_{i+1}^k - u_i^k}{h} - g_{i-\frac{1}{2}}^k \frac{u_i^k - u_{i-1}^k}{h}\right)}{\frac{1}{h^2}\left(g_{i+\frac{1}{2}}^k + g_{i-\frac{1}{2}}^k\right)} \tag{4.4}$$

We assume that the vector u^k contains samples of a corresponding function, i. e. $u_i^k = u^k(x_i)$ on an equidistant grid $\Omega_h := \{x_i = ih\}_{i=0,\dots,N-1}$ for all $k \in \mathbb{N}$. The domain of the functions is then the interval $\Omega = [0, (N-1)h]$. In (4.4) we notice that the iterative scheme contains finite differences which can be understood as approximations of spatial derivatives of u^k. Now we assume that u^k and g are sufficiently smooth to perform a Taylor expansion. For example, we have the term

$$\frac{u_{i+1}^k - u_i^k}{h} = \partial_x u^k \left(x_i + \frac{h}{2}\right) + \mathcal{O}(h^2) \tag{4.5}$$

in the scheme. Together with the abbreviations introduced above this yields

$$g_{i+\frac{1}{2}}^k + g_{i-\frac{1}{2}}^k = g\left(\left|\frac{u_{i+1}^k - u_i^k}{h}\right|^2\right) + g\left(\left|\frac{u_i^k - u_{i-1}^k}{h}\right|^2\right) = 2g\left(|\partial_x u^k(x_i)|^2\right) + \mathcal{O}(h^2) \tag{4.6}$$

and thus we can write

$$u_i^{k+1} = u_i^k + \frac{\partial_x\left(g(|\partial_x u^k(x_i)|^2)\partial_x u^k(x_i)\right) + \mathcal{O}(h^2)}{\frac{1}{h^2}\left(2g(|\partial_x u^k(x_i)|^2) + \mathcal{O}(h^2)\right)} \ . \tag{4.7}$$

To understand the iteration indices $k+1$ and k as discrete samples of a continuous time variable t we introduce a temporal step size $\tau > 0$. Division of both sides by τ leads to the equation

$$\frac{u_i^{k+1} - u_i^k}{\tau} = \frac{\partial_x\left(g(|\partial_x u^k(x_i)|^2)\partial_x u^k(x_i)\right) + \mathcal{O}(h^2)}{\frac{\tau}{h^2}\left(2g(|\partial_x u^k(x_i)|^2) + \mathcal{O}(h^2)\right)} \tag{4.8}$$

where the left-hand side is an approximation for the temporal derivative $\partial_t u$ at time $t = k\tau$ with an error in the order $\mathcal{O}(\tau)$. We set the ratio between h and τ such that $\frac{\tau}{h^2} = \frac{1}{2}$ and let h tend to zero. Then (4.8) approximates

$$\partial_t u = \frac{1}{g(|\partial_x u|^2)}\,\partial_x\left(g(|\partial_x u|^2)\,\partial_x u\right) \tag{4.9}$$

with an error in the order of $\mathcal{O}(\tau + h^2)$. The mirroring boundary conditions are the classical homogeneous Neumann boundary conditions $\partial_x u(x) = 0$.

Interpretation as Accelerated Perona-Malik Filter

Equation (4.9) is similar to the nonlinear diffusion equation presented by Perona and Malik [154]:

$$\partial_t u = \partial_x\left(g(|\partial_x u|^2)\,\partial_x u\right) \ . \tag{4.10}$$

The only difference is the factor $\frac{1}{g(|\partial_x u|^2)}$ on the right-hand side which acts as an acceleration of the Perona-Malik filtering process at edges. To understand this, assume that $|\partial_x u|$ is relatively small within a region. A classical Perona-Malik diffusivity is close to 1 in this case, and the factor has only a small effect. More interesting is the situation near an edge where $\partial_x u$ has large absolute value, and backward diffusion can occur for the diffusivities presented by Perona and Malik. In this case, $g(|\partial_x u|^2)$ is close to zero, and thus $\frac{1}{g(|\partial_x u|^2)}$ leads to an amplification of the backward diffusion behaviour. We can expect such equations to yield sharper results than classical Perona-Malik PDEs. One the other hand, they do not necessarily preserve the average grey value, since they can not be written in divergence form.

4.1.2 Discretisation and Properties

First we take a look at possible discretisations of the scaling limit (4.9) and its stability properties. Then we discuss some problems with leaving the central pixel away in the average and how to overcome these.

Explicit Discretisation

Since classical diffusivities g may be arbitrary close to zero, the fraction $\frac{1}{g(|\partial_x u|^2)}$ in (4.9) is not bounded. This might give rise to concerns regarding stability. However, the weighted averging scheme (4.1) inspires also ways how to obtain stable discretisations: An explicit Euler scheme for (4.9) can be written as

$$u_i^{k+1} \;=\; u_i^k + \tau \,\frac{2}{g_{i+\frac{1}{2}}^k + g_{i-\frac{1}{2}}^k}\,\frac{1}{h}\left(g_{i+\frac{1}{2}}^k \frac{u_{i+1}^k - u_i^k}{h} - g_{i-\frac{1}{2}}^k \frac{u_i^k - u_{i-1}^k}{h}\right) \quad . \tag{4.11}$$

To discuss some properties of this scheme we rewrite the equation as matrix-vector multiplication

$$u^{k+1} \;=\; Q(u^k)u^k \tag{4.12}$$

following the notation in [198]. The matrix $Q(u^k)$ is tridiagonal with the entries

$$Q_{ij}(u^k) \;=\; \begin{cases} \frac{2\tau}{h^2}\,\frac{g_{i+\frac{1}{2}}^k}{g_{i+\frac{1}{2}}^k + g_{i-\frac{1}{2}}^k} & \text{if } j = i+1 \\ 1 - \frac{2\tau}{h^2} & \text{if } j = i \\ \frac{2\tau}{h^2}\,\frac{g_{i-\frac{1}{2}}^k}{g_{i+\frac{1}{2}}^k + g_{i-\frac{1}{2}}^k} & \text{if } j = i-1 \\ 0 & \text{else} \end{cases} \tag{4.13}$$

for inner pixels $i, j \in \{1, \ldots, N-2\}$. At the boundary, we have to be aware of the above mentionned mirroring boundary conditions.

Lemma 4.1 (Stability)
The scheme (4.11) satisfies a maximum-minimum principle if $\tau \leq \frac{h^2}{2}$.

Proof: We see that the matrix $Q(u^k)$ has row sum 1. For $\tau \leq \frac{h^2}{2}$ all three factors are nonnegative, and thus u_i^{k+1} is a convex combination of the three pixels: the scheme is maximum-minimum-stable. □

Further we see that for the limit $\tau = \frac{h^2}{2}$ we obtain exactly the averaging filter (4.1). It should be noted that the stability of our scheme is a consequence of the arithmetic mean used in the fraction in (4.11) to approximate the diffusivity at the position of the pixel x_i.

Remark 4.2 (Matrix Properties)
We have already mentioned above that (4.9) in general does not preserve the average grey value. This is also reflected in the discretisation (4.11). The division by values of g leads to the facts that the matrix is not doubly stochastic [100] as we know it from classical discrete nonlinear diffusion. In fact, it does not have column sum 1 which means that there is in general no redistribution property [198]. Thus it can not be guaranteed for such filters that the average grey value does not change during filtering. However, experiments show that the changes of the average grey value are usually not visible at all.

Involving the Central Pixel in the Averaging Process

The filter (4.1) does not involve the central pixel u_i itself in the average. This might cause problems for certain initial signals: If we choose f to be an alternating signal with two different values, then applying the filter will simply exchange the grey values. To avoid this problem one can give the central pixel a nonnegative weight and involve it in the averaging process. For example, such a modified scheme looks like

$$u_i^{k+1} = \frac{g\left(\left|\frac{u_{i+1}^k - u_i^k}{h}\right|^2\right) u_{i+1}^k + \alpha u_i^k + g\left(\left|\frac{u_{i-1}^k - u_i^k}{h}\right|^2\right) u_{i-1}^k}{g\left(\left|\frac{u_{i+1}^k - u_i^k}{h}\right|^2\right) + \alpha + g\left(\left|\frac{u_{i-1}^k - u_i^k}{h}\right|^2\right)} \tag{4.14}$$

where we have given the central pixel a fixed weight $\alpha > 0$. The same reasoning as presented above relates this averaging filter to the PDE

$$\partial_t u = \frac{1}{\frac{\alpha}{2} + g(|\partial_x u|^2)} \partial_x \Big(g(|\partial_x u|^2)\, \partial_x u \Big) \ . \tag{4.15}$$

Here we see that there is still some factor influencing the velocity of the diffusion process, but this factor now is bounded from above to $\frac{2}{\alpha}$. Compared to (4.9), this slows down the evolution in regions with small derivatives of u.

4.2 Averaging Filters and Scaling Limits in 2-D

In this section we describe two possible approaches to carry over the one-dimensional results obtained so far to the two-dimensional setting.

4.2.1 Local Averaging and Anisotropic Diffusion

As direct two-dimensional analogue to (4.1), we start with weighted averaging over the direct neighbouring pixels. Let $\mathcal{N}(i)$ be the set of indices of the maximal four direct neighbours of the pixel with index i. Then an equivalent of the weighted averaging filter (4.1) in two dimensions can be written as

$$u_i^{k+1} \;=\; \frac{\sum\limits_{j\in\mathcal{N}(i)} g\left(\left|\frac{u_j^k-u_i^k}{h}\right|^2\right) u_j^k}{\sum\limits_{j\in\mathcal{N}(i)} g\left(\left|\frac{u_j^k-u_i^k}{h}\right|^2\right)} \;. \tag{4.16}$$

Numerator and denominator of this scheme can be understood as the sum of numerators and denominators of two one-dimensional schemes in x- and y-direction. Thus the reasoning described in the last section shows that (4.16) is a consistent approximation for

$$\partial_t u \;=\; \frac{\partial_x\left(g(|\partial_x u|^2)\partial_x u\right) + \partial_y\left(g(|\partial_y u|^2)\partial_y u\right)}{g(|\partial_x u|^2) + g(|\partial_y u|^2)} \;. \tag{4.17}$$

This equation is not rotationally invariant, and thus will lead to artefacts in images with rotational invariant objects. This indicates that also the weighted averaging method (4.16) leads to such artefacts which is shown with a practical example in Figure 4.2.

To circumvent this shortcoming, we understand (4.17) as a crude approximation of the rotationally invariant equation

$$\partial_t u \;=\; \frac{1}{\int\limits_0^\pi g(|\partial_{e_\varphi} u|^2)\,d\varphi} \cdot \int\limits_0^\pi \partial_{e_\varphi}\left(g(|\partial_{e_\varphi} u|^2)\partial_{e_\varphi} u\right) d\varphi \tag{4.18}$$

where we write $e_\varphi = (\cos(\varphi), \sin(\varphi))^T$ for the unit vector in direction φ. In (4.17) the integrals are approximated as trapez sums where only two evaluation points of the integrands are used. Similar to Catté et al. [32] we introduce a smoothing of the argument of the diffusivity by the convolution of u with a Gaussian kernel of standard deviation σ, and we write $u_\sigma := G_\sigma * u$. This convolution can also simply be introduced in the arguments of the weights used in the averaging process (4.16). It does not affect the reasoning leading to the PDE (4.18).

An equation similar to (4.18) has been studied in [196] in the context of anisotropic diffusion filtering:

$$\partial_t u \;=\; \frac{2}{\pi} \int_0^\pi \partial_{e_\varphi} \Big(g(|\partial_{e_\varphi} u_\sigma|^2) \partial_{e_\varphi} u \Big)\, d\varphi. \tag{4.19}$$

This equation has been called the *integration model for anisotropic diffusion* [197]. The proofs in [196] can be applied to show that (4.18) can be transformed into

$$\partial_t u \;=\; \frac{1}{\mathrm{trace}(D(\nabla u_\sigma))}\, \mathrm{div}\Big(D(\nabla u_\sigma) \cdot \nabla u \Big) \tag{4.20}$$

with the diffusion tensor

$$D(\nabla u_\sigma) \;=\; \int_0^\pi e_\varphi e_\varphi^\top\, g(|\partial_{e_\varphi} u_\sigma|^2)\, d\varphi. \tag{4.21}$$

In [196] the eigenvectors of this diffusion tensor $D(\nabla u_\sigma)$ are calculated as

$$v_1(\psi) \;=\; \begin{pmatrix} -\sin(\psi) \\ \cos(\psi) \end{pmatrix} \quad \text{and} \quad v_2(\psi) \;=\; \begin{pmatrix} \cos(\psi) \\ \sin(\psi) \end{pmatrix} \tag{4.22}$$

where $\nabla u_\sigma \neq 0$ and (r, ψ) are the polar coordinates of ∇u_σ. That means v_1 is the direction of the isophote of u_σ (along an edge), while v_2 is the direction across the edge. The corresponding eigenvalues are given by

$$\lambda_1(\nabla u_\sigma) \;=\; \int_0^\pi \sin^2(\varphi)\, g(|\partial_{e_\varphi} u_\sigma|^2)\, d\varphi \quad \text{and} \tag{4.23}$$

$$\lambda_2(\nabla u_\sigma) \;=\; \int_0^\pi \cos^2(\varphi)\, g(|\partial_{e_\varphi} u_\sigma|^2)\, d\varphi \;\; . \tag{4.24}$$

Equation (4.20) is the relevant formulation for practical implementations. This equation is rotationally invariant, since the eigenvectors follow a rotation of the input image, and the eigenvalues are invariant under image rotations. Since

$$\mathrm{trace}(D(\nabla u_\sigma)) \;=\; \int_0^\pi g(|\partial_{e_\varphi} u_\sigma|^2)\, d\varphi$$

is always significantly larger than zero, the sharpening of the edges will be less pronounced in this anisotropic case than in the one-dimensional case (4.9). This effect is similar to the one described at the example of (4.15). Nevertheless, we are going to see with numerical examples that not only preservation of edges, but also sharpening is possible with this filter.

4.2.2 Rotational Invariance with the Bilateral Filter

In the last subsection we have derived an anisotropic PDE filter from a weighted averaging of the direct neighbouring pixels. To circumvent the lack of rotational invariance in (4.17) we have understood it as a very crude approximation of the rotational invariant approach (4.18). Nevertheless, there are discrete filters which address the problem of lacking rotational invariance by involving information from pixels in a larger neighborhood.

We consider here the prominent example of the bilateral filter [185, 11, 72]. Even though this filter is proposed as a noniterative method, it can make sense to perform several filtering steps; thus we understand it as an iterative averaging filter. In one filtering step, not only the direct neighbouring pixels are involved in the averaging, but an extended neighbourhood $i + \mathcal{B}_R$. Here

$$\mathcal{B}_R \;=\; \{j \in \mathbb{R}^2 : |j| \le R\} \cap \Omega_h \tag{4.25}$$

denotes the intersection of the disc of radius R in $\mathbb{R}^2$ with the pixel grid Ω_h. A variant of the bilateral filter then looks like this:

$$u_i^{k+1} \;=\; \frac{\sum\limits_{j\in\mathcal{B}_R} g\left(\frac{|u_{i+j}^k - u_i^k|^2}{|j|^2}\right)\frac{w(|j|)}{|j|^2}u_{i+j}^k}{\sum\limits_{j\in\mathcal{B}_R} g\left(\frac{|u_{i+j}^k - u_i^k|^2}{|j|^2}\right)\frac{w(|j|)}{|j|^2}} \;. \tag{4.26}$$

The spatial distance between x_i and x_{i+j} results in a usually smaller weight $w(|j|)/|j|^2$, where an example for w is $w(h) = h^2\exp(-h^2)$. In this special example, w leads to a Gaussian weight depending on the distance of the two pixels.

We now want to imitate the approach presented in Section 4.1. To this end we only consider one half of the disc $\mathcal{B}_R^+ = \{(x,y) \in \mathcal{B}_R | x \ge 0\}$ and rewrite the sums in (4.26) as

$$u_i^{k+1} \;=\; \frac{\sum\limits_{j\in\mathcal{B}_R^+} \frac{w(|j|)}{|j|^2}\left(g\left(\frac{|u_{i+j}^k - u_i^k|^2}{|j|^2}\right)u_{i+j}^k + g\left(\frac{|u_i^k - u_{i-j}^k|^2}{|j|^2}\right)u_{i-j}^k\right)}{\sum\limits_{j\in\mathcal{B}_R^+} \frac{w(|j|)}{|j|^2}\left(g\left(\frac{|u_{i+j}^k - u_i^k|^2}{|j|^2}\right) + g\left(\frac{|u_i^k - u_{i-j}^k|^2}{|j|^2}\right)\right)} \;. \tag{4.27}$$

The novelty in this two-dimensional case is that we have to consider several directional derivatives. We see that there appear directional finite differences in (4.27). Let $e_\varphi = \frac{j}{|j|}$ be the unit vector pointing in the direction of $j \neq 0$, and $h = |j|$ be the length of the vector j. A Taylor expansion of u around the pixel i yields

$$u(x_{i+j}) \;=\; u(x_i) + \langle \nabla u, j\rangle + \mathcal{O}(h^2) \;=\; u_i + (\partial_{e_\varphi} u)\cdot h + \mathcal{O}(h^2)$$

which will be useful in the form

$$\frac{u_{i+j} - u_i}{h} \;=\; \partial_{e_\varphi} u + \mathcal{O}(h) \;. \tag{4.28}$$

Applying the Taylor formula (4.28) to (4.27) allows us to write

$$u^{k+1}(x_i) - u^k(x_i) \;=\; \frac{\sum\limits_{j\in\mathcal{B}_R^+} w(h)\,\left(\partial_{e_\varphi}\left(g(|\partial_{e_\varphi}u|^2)\partial_{e_\varphi}u\right) + \mathcal{O}(h^2)\right)}{\sum\limits_{j\in\mathcal{B}_R^+} \frac{2w(h)}{h^2}\,\left(g(|\partial_{e_\varphi}u|^2) + \mathcal{O}(h^2)\right)} \tag{4.29}$$

At this point we investigate the scaling limit if we let the spatial step sizes in x- and y-direction tend to zero while we keep the size R of the neighbourhood fixed. This means that the number of grid points in our neighbourhood $\mathcal{B}_R$ is tending to infinity. Thus we can consider the sums in (4.29) as Riemann sums which approximate integrals over the set $\mathcal{B}_R^+$. We notice that the arguments in (4.29) are the length of j and its angle which immediately suggests to write the integrals in polar coordinates:

$$u^{k+1} - u^k = \frac{1}{\int\limits_0^R \frac{2w(h)}{h^2} \int\limits_0^\pi g(|\partial_{e_\varphi}u^k|^2)\,d\varphi\,dh} \cdot \int\limits_0^R w(h) \int\limits_0^\pi \partial_{e_\varphi}\left(g(|\partial_{e_\varphi}u^k|^2)\partial_{e_\varphi}u^k\right) d\varphi dh \tag{4.30}$$

Since the inner integrals do not depend on the radius r, the outer ones are just a scaling factor, that means (4.30) is corresponding up to a constant factor to

$$u^{k+1} - u^k \;=\; \frac{1}{\int\limits_0^\pi g(|\partial_{e_\varphi}u^k|^2)\,d\varphi} \cdot \int\limits_0^\pi \partial_{e_\varphi}\left(g(|\partial_{e_\varphi}u^k|^2)\partial_{e_\varphi}u^k\right)\,d\varphi \;. \tag{4.31}$$

If we understand the right-hand side as temporal forward difference we can see (4.31) as an approximation to (4.18). This provides an interpretation of bilateral filtering as an anisotropic PDE.

Remark 4.3 (Redundant Derivative Approximation)
Involving different points at the same direction φ with different radius r is not necessary in the continuous setting, but makes sense for a discrete formulation: For example, with the help of the grey value u_{i+j} we obtain one approximation for the directional derivative $\partial_{e_\varphi}u$, and u_{i+2j} yields another one. The larger neighbourhood now allows to incorporate both of them while the scaling with $w(2r)$ results in a smaller weight for the approximation with higher error.

Finally we want to point out that the underlying ideas in this section are not restricted to the two-dimensional case, but can also be carried over to higher dimensions.

4.3 Generalised Mean Curvature Motion

There is no unique way to generalise (4.9) to two or more dimensions. As described in the last section, considering a 2-D weighted averaging scheme as starting point and taking the scaling limit leads to an anisotropic diffusion equation including a diffusion tensor.

In this section, we generalise (4.9) in a different way to 2-D: We replace the first derivative of u in (4.9) by a gradient and the outer derivative by the divergence. This directly leads to the PDE

$$\partial_t u \;=\; \frac{1}{g(|\nabla u|^2)}\,\mathrm{div}\Big(g(|\nabla u|^2)\nabla u\Big)\;, \tag{4.32}$$

which will be called *generalised mean curvature motion (GMCM)* here. To justify this name, consider the special case $g(s^2) = 1/s$ to obtain the standard mean curvature motion equation (1.31). This already indicates that the additional factor on the right-hand side changes the behaviour compared to the Perona-Malik model more than in the 1-D case (4.9).

4.3.1 Interpretation

From the decomposition of the Perona-Malik filter in (1.28), we immediately derive that generalised mean curvature motion (4.32) can be decomposed as

$$\begin{aligned}\partial_t u &= u_{\xi\xi} + \left(1 + 2\,\frac{g'(|\nabla u|^2)|\nabla u|^2}{g(|\nabla u|^2)}\right) u_{\eta\eta} &(4.33)\\ &= u_{\xi\xi} + a(|\nabla u|^2)\,u_{\eta\eta}\;. &(4.34)\end{aligned}$$

That means we have a mean curvature motion equation with an additional diffusive component in orthogonal direction η which is steered by the factor

$$a(s^2) \;:=\; 1 + 2\,\frac{g'(s^2)s^2}{g(s^2)}\;. \tag{4.35}$$

As argument s^2, the factor depends on the squared norm of the gradient $|\nabla u|^2$. We will discuss later how the choice of g influences the behaviour of this factor $a(s^2)$. The basic idea is that the filter performs shrinkage of level lines in the sense of mean curvature motion while the second term keeps edges sharp during the evolution.

There is also another way of understanding the process: Having the equation $\Delta u = u_{\xi\xi} + u_{\eta\eta}$ in mind, we can rewrite this as

$$\partial_t u \;=\; \Delta u + 2\,\frac{g'(|\nabla u|^2)|\nabla u|^2}{g(|\nabla u|^2)}\,u_{\eta\eta}\;. \tag{4.36}$$

In this form, we see that the generalised mean curvature motion can be understood as linear diffusion with an additional shock term for edge enhancement. For more information on shock filters, see [149, 94, 201]. While classical Perona-Malik filtering slows down the linear diffusion part near edges by the factor $g(|\nabla u|)$, the velocity of this part is constant for generalised mean curvature motion.

4.3.2 Choices for the Nonlinearity

After specifying the general framework, we now focus on several choices of the function g and give some first insight in the expected behaviour of the corresponding methods in terms of forward and backward diffusion (similar to Section 2.2.3).

Example 4.4 (Perona-Malik Diffusivity)
Let us choose the classical diffusivity function $g(s^2) = (1 + s^2/\lambda^2)^{-1}$ proposed by Perona and Malik [154]. This diffusivity is especially interesting because it is capable of switching between forward and backward diffusion adaptively. In this case we have

$$a(s^2) \;=\; 1 + \frac{g'(s^2)s^2}{g(s^2)} \;=\; 1 - 2\frac{s^2}{s^2+\lambda^2} \tag{4.37}$$

which immediately shows that $-1 \leq a(s^2) \leq 1$ for all $s \in \mathbb{R}$. In a region where $|\nabla u| < \lambda$, we have forward diffusion. That means the whole process (4.33) acts like linear diffusion there. Close to an edge, we have forward diffusion along the edge and backward diffusion across the edge. This explains the edge-preserving behaviour which can be observed at the practical results in Section 4.4.

Example 4.5 (Unbounded Backward Diffusion)
Another frequently used diffusivity function is $g(s^2) = \exp\left(-s^2/(2\lambda^2)\right)$ which has also been proposed by Perona and Malik [154]. In the classical nonlinear diffusion approach, it has the same properties as the function discussed above. In our case we obtain $a(s^2) = 1 - \frac{s^2}{\lambda^2}$. We have $a(s^2) \leq 1$ for all $s \in \mathbb{R}$, but a is not bounded from below. That means in theory there would be no limit for the amount of backward diffusion in a pixel where $|\nabla u|$ is very large. We see that similar diffusion properties do not imply a similar behaviour in the corresponding GMCM model. Nevertheless, this special example is of rather theoretical interest, since for realistic values of $|\nabla u|$ and λ, the values $\exp\left(|\nabla u|^2/(2\lambda^2)\right)$ and $\exp\left(-|\nabla u|^2/(2\lambda^2)\right)$ differ by so many orders of magnitude that even a representation in `double` precision on a standard PC is not precise enough for an implementation.

Example 4.6 (Constant Diffusion Velocity in Both Directions)
So far, we have chosen the amount of diffusion in direction η adaptively depending

on the gradient magnitude of the evolving image $|\nabla u|$. Now we consider the case that the diffusion in direction η has a constant velocity. This is equivalent to

$$a(s^2) \;=\; 1 + \frac{g'(s^2)s^2}{g(s^2)} \;=\; c \in \mathbb{R} \;. \tag{4.38}$$

We see that this condition is satisfied for the family of functions $g(s^2) = \frac{1}{|s|^p}$ for $p > 0$ where we have $a(s^2) = 1 - p$. The corresponding equation is given by

$$\partial_t u \;=\; |\nabla u|^p \operatorname{div}\left(\frac{\nabla u}{|\nabla u|^p}\right) \;=\; u_{\xi\xi} + (1-p)\, u_{\eta\eta} \;. \tag{4.39}$$

For $p = 1$, we obtain mean curvature motion (1.31) as special case of this equation while $p = 0$ yields the linear diffusion equation.

The filter (4.39) has some similarities with an evolution equation using the p-Laplacian [115, 12]:

$$\begin{aligned} \partial_t u &= \Delta_p u \\ &= \operatorname{div}\Big(|\nabla u|^{p-2}\, \nabla u\Big) \\ &= |\nabla u|^{p-2}\, (u_{\xi\xi} + (p-1) u_{\eta\eta}) \;. \end{aligned} \tag{4.40}$$

This equation can be motivated as gradient descent equation for the energy

$$\mathscr{E}(u) \;=\; \frac{1}{p} \int_\Omega |\nabla u|^p dx \;. \tag{4.41}$$

In the field of image processing, it has first been proposed for image interpolation by Caselles et al. [31]. It has also been used for the evolution of shapes [47]. Recently, it has been proposed to use it for image smoothing by Kuijper [113]. For this equation, $p = 2$ yields linear diffusion, and $p = 1$ gives us the total variation flow.

In the experimental section, we are going to take a closer look at the behaviour of the filter family (4.39) for several values of p.

Remark 4.7 (Historical Background)
A certain special case of this family of methods has been proposed already in 1965 by Gabor in the context of electron microscopy [82]. Later on, his approach has been reviewed and brought into the context of contemporary image analysis by Lindenbaum et al. [119]. Rewriting his approach in our notation gives the equation

$$u \;=\; f - \frac{\mu^2}{2}\left(f_{\eta\eta} - \frac{1}{3} f_{\xi\xi}\right) \tag{4.42}$$

for an initial image f and a filtered version u. The quantity μ is derived from the application. We rewrite this equation as

$$\frac{6}{\mu^2}(u - f) \;=\; f_{\xi\xi} - 3 f_{\eta\eta} \;. \tag{4.43}$$

The left-hand side of (4.43) can be interpreted as finite difference approximation of a time-derivative. That means, Gabor's approach (4.42) can be seen as one step in an Euler forward time-discretisation of (4.39) with $p = 4$ and time step size $\tau = \frac{\mu^2}{6}$. Due to limited computational tools, the approach was rather theoretically motivated than used in practice at that time.

4.3.3 Discretisation

So far, we have not addressed the problem of discretising generalised mean curvature motion (4.33). In our first practical examples, it turns out that the question of finding a suitable discretisation is very important for this kind of equations.

Let $h_1, h_2 > 0$ be the pixel distance in x- and y-direction and $\mathcal{N}_d(i)$ the indices of the direct neighbouring pixels in direction d to the pixel with index i. Let u_i^k denote the grey value of pixel i at the time step k. We start with the grey values of the given initial image $u_i^0 := f_i$. Let further $g_i^k \approx g(|\nabla u(x_i)|^2)$ denote an approximation to the weighting function evaluated at pixel i. We have used the approximation

$$|\nabla u(x_i)|^2 \;\approx\; \sum_{d=1}^{2} \sum_{j \in \mathcal{N}_d(i)} \frac{(u_j - u_i)^2}{2h_d^2} \;. \tag{4.44}$$

This approximation yields better results for (4.33) than standard central differences. Similar to [200] we use a finite difference scheme with harmonic averaging of the diffusivity approximations:

$$u_i^{k+1} \;=\; \begin{cases} u_i^k \;, & \text{if } g_i^k = 0 \\ u_i^k + \dfrac{\tau}{g_i^k} \displaystyle\sum_{d=1}^{2} \sum_{j \in \mathcal{N}_d(i)} \frac{2}{\frac{1}{g_j^k} + \frac{1}{g_i^k}} \frac{u_j^k - u_i^k}{h_d^2} \;, & \text{else .} \end{cases} \tag{4.45}$$

Why this scheme is very stable in practice can be seen by a simple equivalent reformulation of the scheme for $g_i^k \neq 0$:

$$u_i^{k+1} \;=\; u_i^k + \tau \sum_{d=1}^{2} \sum_{j \in \mathcal{N}_d(i)} \frac{2g_j^k}{g_j^k + g_i^k} \frac{u_j^k - u_i^k}{h_d^2} \;. \tag{4.46}$$

Under the assumption that g is a non-negative function, we have $0 \leq \frac{g_j^k}{g_j^k + g_i^k} \leq 1$. It follows that

$$\sum_{d=1}^{2} \sum_{j \in \mathcal{N}_d(i)} \frac{2g_j^k}{(g_j^k + g_i^k)h_d^2} \;\leq\; \sum_{d=1}^{2} |\mathcal{N}_d(i)| \frac{2}{h_d^2} \;.$$

Since we have at most 2 direct neighbours in each direction, we can write

$$\sum_{d=1}^{2} \sum_{j \in \mathcal{N}_d(i)} \frac{2g_j^k}{(g_j^k + g_i^k)h_d^2} \;\leq\; \frac{4}{h_1^2} + \frac{4}{h_2^2} \;.$$

This gives

$$1 \;\geq\; 1 - \tau \sum_{d=1}^{2} \sum_{j \in \mathcal{N}_d(i)} \frac{2g_j^k}{(g_j^k + g_i^k)h_d^2} \;\geq\; 1 - \tau \left(\frac{4}{h_1^2} + \frac{4}{h_2^2} \right) \;,$$

and thus the scheme is a convex combination of grey values from the previous step for

$$\tau \;\leq\; \left(\frac{4}{h_1^2} + \frac{4}{h_2^2} \right)^{-1} \;=\; \frac{h_1^2 h_2^2}{4h_1^2 + 4h_2^2} \;.$$

In practice we often have $h_1 = h_2 = 1$, and in this case we have to choose $\tau \leq \frac{1}{8}$. We conclude that by a suitable choice of τ, we can achieve

$$\min_{j \in J} f_j \;\leq\; u_i^k \;\leq\; \max_{j \in J} f_j \tag{4.47}$$

for all $k \in \mathbb{N}$ and all $i \in J$, i. e. the process satisfies a maximum-minimum principle.

4.4 Numerical Experiments

Now we show some numerical examples to illustrate the practical behaviour of averaging methods and our novel PDE methods. As weight function or diffusivity we use the classical diffusivity $g(s) = (1 + s^2/\lambda^2)^{-1}$ by Perona and Malik [154].

First we display an experiment in the one-dimensional case in Figure 4.1. We see that the presence of the acceleration factor allows for sharper edges. With the same evolution time we can achieve a stronger edge enhancement than with a classical nonlinear diffusion equation of Perona-Malik type. Figure 4.2 visualises the lack of rotational invariance of local averaging filters and how it can be improved with a larger neighbourhood in the bilateral filter. Even a better effect than extending the neighbourhood can be achieved with the anisotropic

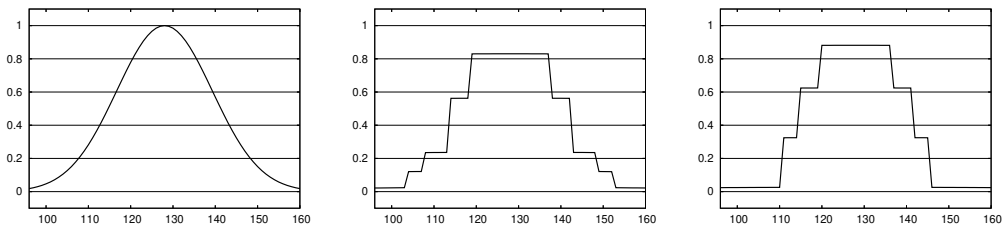


Figure 4.1: Accelerated Perona-Malik diffusion in 1-D. *Left:* Original signal (64 point width section of a signal with 256 pixels). *Middle:* Perona-Malik diffusion ($\lambda = 0.005$, $t = 5000$). *Right:* Perona-Malik diffusion with additional factor (4.9) and the same parameters.

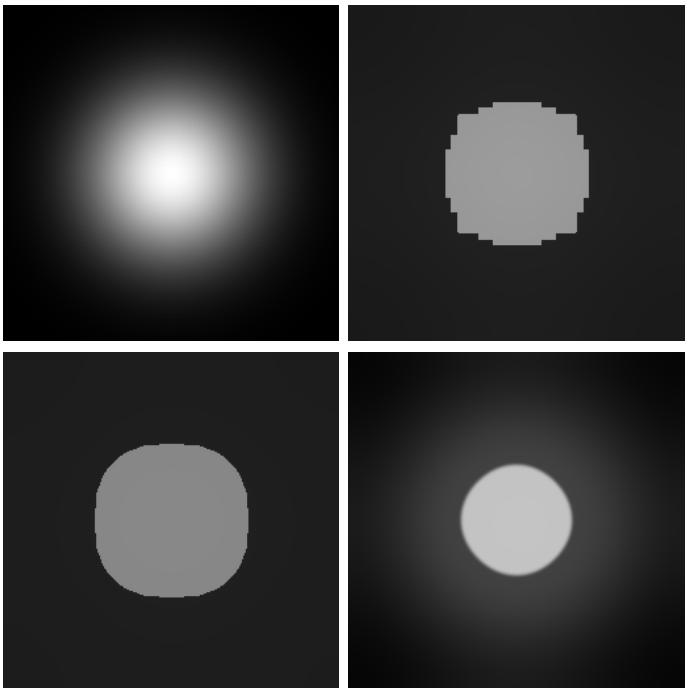


Figure 4.2: Weighted averaging and accelerated diffusion. *Top left:* Original image (size: 256×256 pixels). *Top right:* Weighted averaging (equation (4.16), $\lambda = 3.0$, 15000 iterations). *Bottom left:* Iterated bilateral filtering ($\lambda = 3.0$, window size 5×5 pixels, $w(h) = h^2 \exp(-h^2/4)$, 5000 iterations). *Bottom right:* Accelerated anisotropic diffusion ($\lambda = 10$, $\sigma = 2$, $t = 1660$).

nonlinear diffusion equation (4.18). Figure 4.3 shows the denoising capabilities of the anisotropic diffusion equation (4.18) for real-world data. The anisotropic behaviour is clearly visible.

Now we study the properties of generalised mean curvature motion with some practical examples. In Figure 4.4 we compare the results of Perona-Malik filtering with mean curvature motion and generalised mean curvature motion. It can be seen that GMCM offers a combination of the properties of MCM with Perona-Malik filtering: On the one hand, the contrast parameter λ gives us the opportunity to distinguish between smoothed and preserved sharp edges as known from Perona-Malik filtering. On the other hand, the objects are shrunken to points and vanish at finite time as known from mean curvature motion.

In our second experiment with GMCM, we compare the behaviour of equation (4.39) for different values of p. The corresponding test images can be seen in Figure 4.5. Figure 4.6 shows the results of the application to the same test image. The implementation for colour images has been done with joint diffusivity following the ideas of Gerig et al. [87]. We see that $p = 1$ yields blurred results while $p \geq 2$ leads to sharp edges. We notice that some basic properties of mean

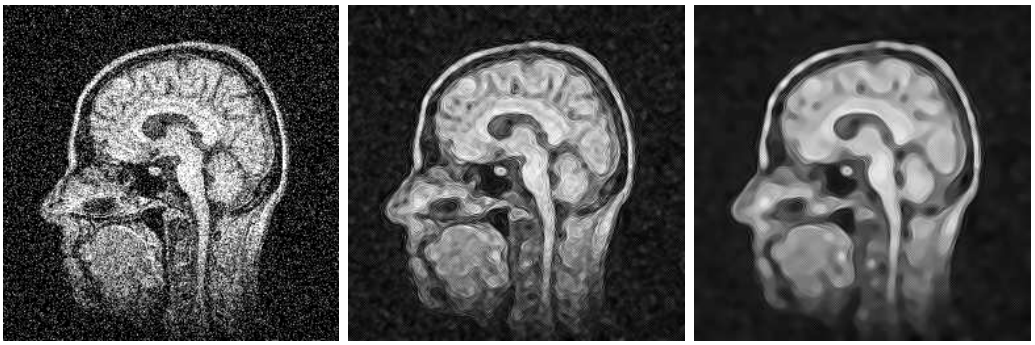


Figure 4.3: Accelerated diffusion. *Left:* Original image (size: 256×256 pixels) and additive Gaussian noise with standard deviation $\sigma = 50$. *Middle:* Accelerated anisotropic diffusion ($\lambda = 2$, $\sigma = 3$, $t = 2$). *Right:* Same, but with $t = 10$.

curvature motion are also satisfied here even if we do not have a theoretical reason for this so far: At this example we see that nonconvex objects are getting convex and shrink in finite time. Further, for larger p it is possible that corners are also kept longer in the iterations, the process of making objects circular is slowed down. That means objects are getting smaller during evolution while the shape is preserved longer than for mean curvature motion.

We have already mentioned that one important property of mean curvature motion is the morphological invariance. We use a test image composed out of four circles with different contrast to the background (see Figure 4.4) to determine the contrast dependence of generalised mean curvature motion (4.39). We see that for $p = 2, 4, 6, 10$ the four circles in one filtering result always have very similar size. This means, for constant regions, the contrast in this example does hardly influence the shrinkage time. We know from Figure 4.6 that these processes tend to segment images into constant regions after a few steps. Further we notice that the stopping times for shrinkage of the circles changes strongly with p. Our

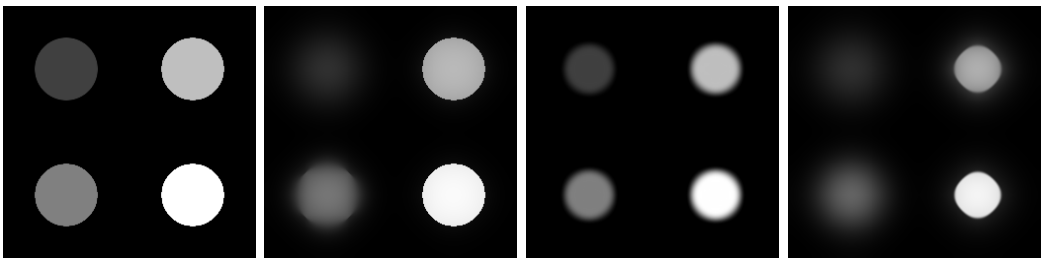


Figure 4.4: Comparison of different filtering techniques. *Left:* Original image, 256×256 pixels. *Second left:* Perona-Malik filtering, $\lambda = 10$. *Second right:* Mean curvature motion. *Right:* Generalised mean curvature motion (4.32) with $g(s) = (1 + s^2/\lambda^2)^{-1}$, $\lambda = 10$. Stopping time in all examples: $t = 200$.

Figure 4.5: Real-world test image. *Left:* Original image, 512×340 pixels. *Right:* With additive Gaussian noise, standard deviation $\sigma = 20$.

experience which is also confirmed by a larger number of experiments is that the stopping time is smallest for $p = 4$ and increases rapidly for larger p. In Figure 4.8, we see how joint denoising and curve shrinking is possible with generalised mean curvature motion. In this example, it is possible to obtain sharp edges even for a highly noisy test image. At the same time, the process shrinks circles with a comparable velocity to mean curvature motion. We see that self-snakes also denoise the image, but do not shrink it even for a larger stopping time.

Our last experiment in this chapter presents a possible application for generalised mean curvature motion: Depending on the contrast, one can distinguished between objects two classes of objects in an image: The ones with high contrast will be shrunken with an evolution similar to mean curvature motion, while the ones with low contrast are smoothed similar to linear diffusion. The corresponding example is displayed in Figure 4.9. We see that with increasing the contrast parameter λ, more objects are smoothed. The evolution for the shrunken objects is not visibly influenced by the larger parameter. For $\lambda = 50$, all objects are smoothed. We have performed the same experiment for an artificial test image and a noisy version of it. We see that the process is highly robust under noise. Only for $\lambda = 1$ there appear some problems in recovering the exakt shape of two of the objects.

4.5 Summary

In this chapter, we have investigated the close relationship between weighted averaging processes and PDE-based filtering methods. Starting from a very simple averaging filter using only the two direct neighbouring pixels in 1-D, we have derived a modified Perona-Malik equation [154]. The modification consists of a factor that can accelerate the sharpening of edges and thus may give an improved edge enhancment. This does not impose problems with stability for suitable discretisations.

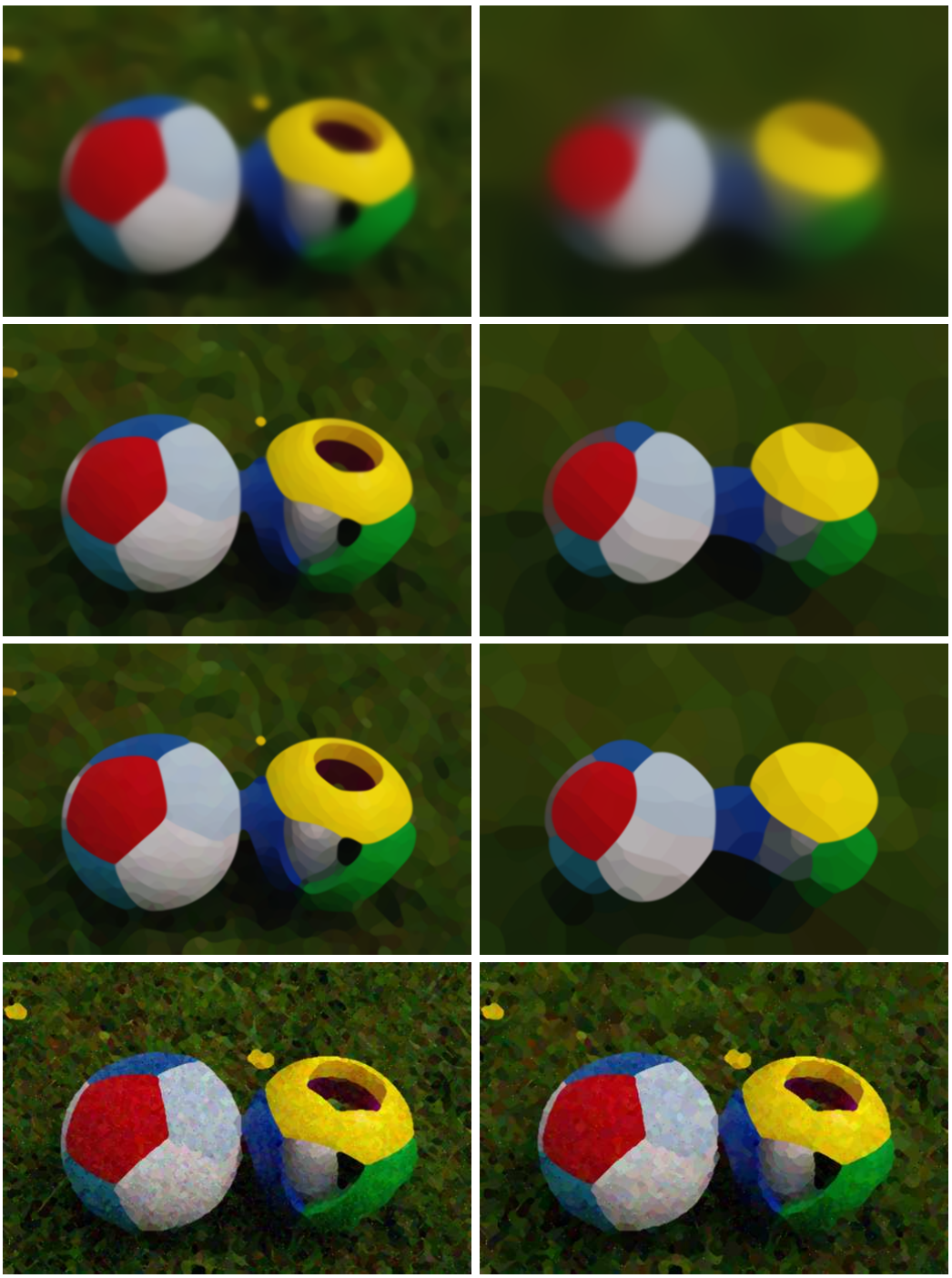


Figure 4.6: Comparison of the evolution (4.39) for different values of p. *Rows from top to bottom:* $p = 1, 1.5, 2, 8$. *Left column:* $t = 100$. *Right column:* $t = 1000$.

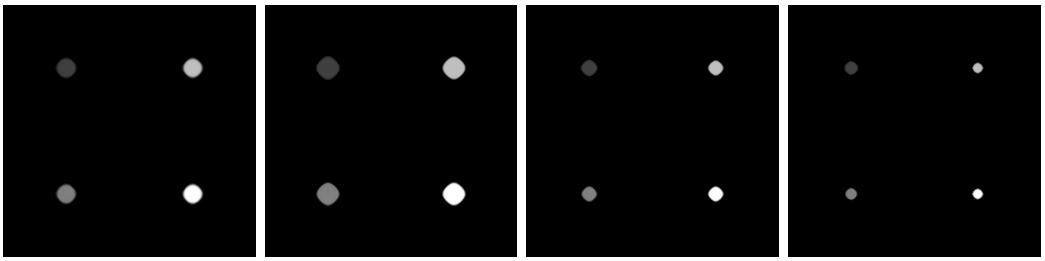


Figure 4.7: Contrast dependency of a discrete version for the constant generalised mean curvature motion (4.39). *Left:* $p = 2, t = 375$. *Second left:* $p = 4, t = 312.5$. *Second right:* $p = 6, t = 375$. *Right:* $p = 10, t = 625$.

We have followed several ways to obtain related equations in the 2-D case: Starting with a simple 2-D averaging filter using only the smallest possible neighbourhood, the four direct neighbouring pixels, can lead to the lack of rotational invariance. However, one can regard the corresponding PDE as a crude approximation of a rotationally invariant method closely related to an anisotropic diffusion filter. The same PDE can also be derived starting from an iterated bilateral filter using all neighbouring pixels inside a neighbourhood of extended size, namely a circle, in the averaging process. This link to anisotropic diffusion helps in the understanding of the widely-used bilateral filter.

Instead of starting with 2-D averaging schemes, it is also possible directly to transfer the 1-D scaling limit to the two-dimensional setting. This leads to a family of partial differential equations comprising mean curvature motion and containing a nonlinearity. Numerical experiments show that the family somehow inherits perperties from both mean curvature motion and Perona-Malik filtering: On the one hand, objects are shrunken to ellipses and vanish after a finite time like it is well-known for mean curvature motion. On the other hand, by choosing the nonlinearity and the scale parameter suitably, it is possible to select edges to be preserved and edges to be smoothed as we know it from Perona-Malik diffusion. It is possible to choose the nonlinearity such that the obtained results are much sharper than with classical mean curvature motion while the interesting properties of this filter are kept.

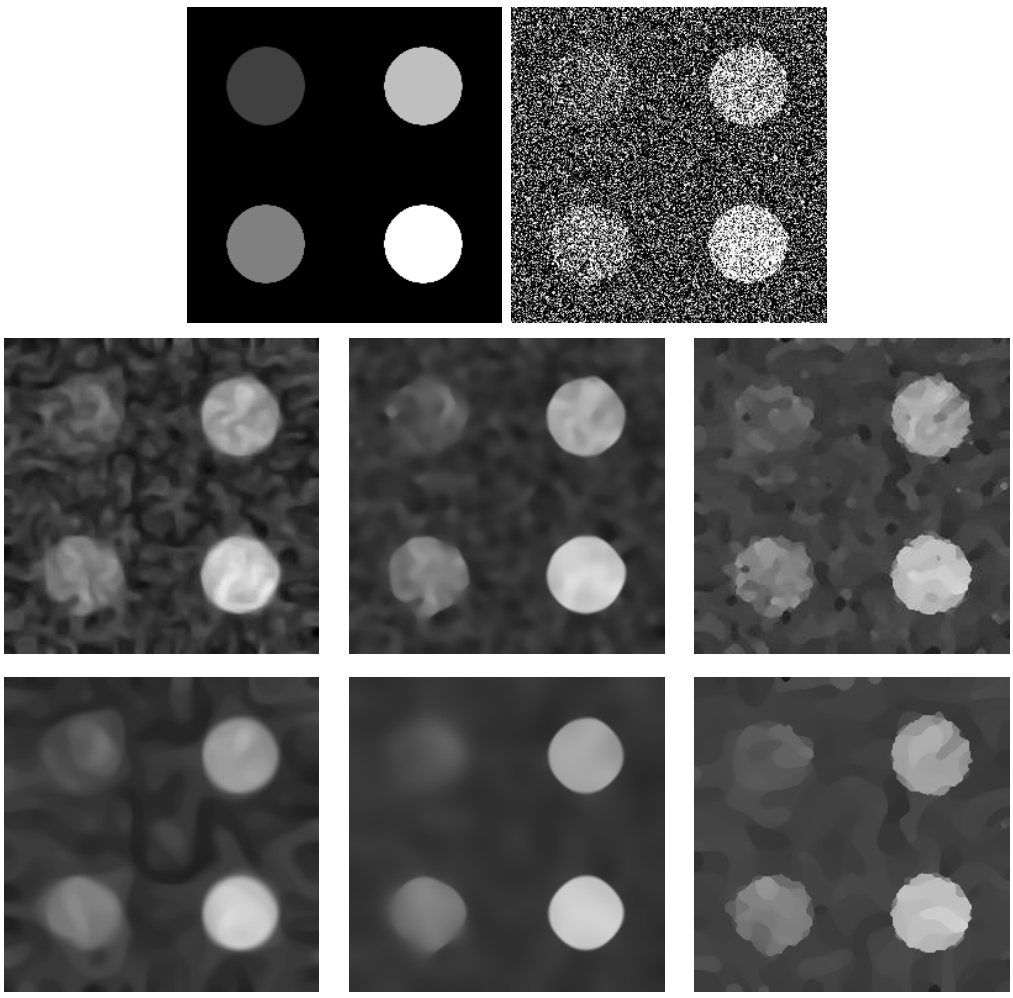


Figure 4.8: Joint denoising and curvature motion. *Top left:* Original image, 256×256 pixels. *Top right:* Original image with Gaussian noise, standard deviation $\sigma = 200$. *First column:* MCM, $t = 12.5, 50$. *Second column:* Generalised mean curvature motion (4.32), $g(s^2) = (1 + s^2/\lambda^2)^{-1}$, $t = 12.5, 50$. *Third column:* Self-snakes, $\lambda = 10, t = 50, 100$.

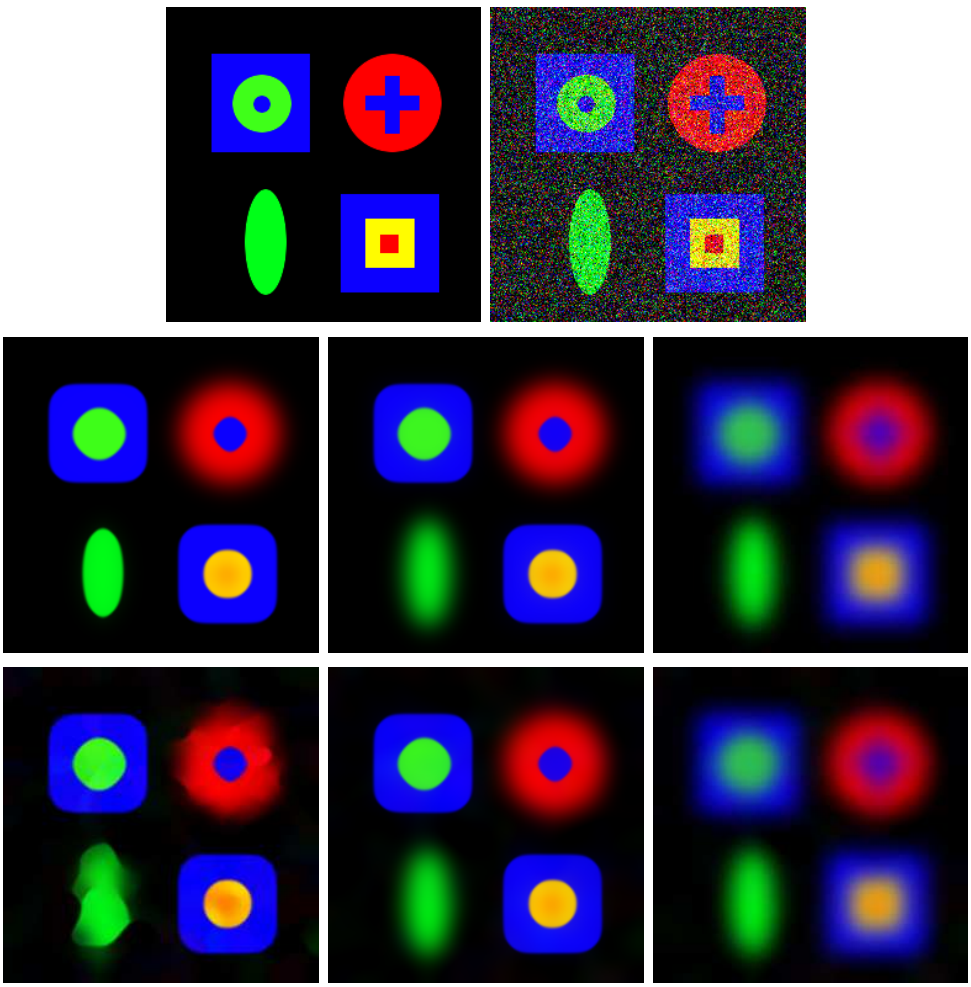


Figure 4.9: Generalised mean curvature motion with $g(s^2) = (1 + s^2/\lambda^2)^{-1}$ and stopping time $t = 50$. *Top left:* Original image, 256×256 pixels. *Top right:* With additive Gaussian noise, standard deviation $\sigma = 100$. *Middle row:* Results for the original image. *Bottom row:* Results for the noisy image. *From left to right:* $\lambda = 1, 10, 50$.

Chapter 5

Semilocal Methods, NDS

This chapter investigates the unifying image simplification approach with nonlocal data and smoothness terms (NDS) which has been introduced by Mrázek et al. in [139] and already been mentioned in Subsection 1.4.2. We focus on characterisation of minimisers for the discrete energy function. Instead of using gradient descent methods with adaptive time step sizes we prefer to formulate the necessary conditions for minimisers as fixed point equations. This gives an alternative motivation for the connection to some classical methods and allows to prove the existence of minimisers. To speed up the convergence we also use two approaches involving Newton's method which is only applicable for strictly convex penalisers. The running time in practice is studied with numerical examples in 1-D and 2-D. A preliminary version of this chapter has been published in [58].

This chapter is organised as follows: Section 5.1 gives a closer description of the energy functional we deal with and its relations to well-known filtering methods like M-smoothers and the bilateral filter. In Section 5.2 we discuss different approaches to minimise the NDS functional including a fixed point scheme and Newton's method. Numerical experiments in 1-D and 2-D in Section 5.3 compare the behaviour and running time of the presented approaches. A summary of the results and an outlook in Section 5.4 conclude the chapter.

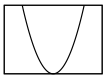
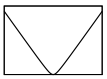
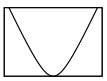
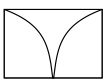
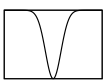
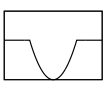
5.1 The NDS Model

We give a short sketch of the discrete NDS energy function and describe how it can be related to a choice of classical image simplification methods.

5.1.1 The NDS Energy Function

Let us first review the NDS model: Let $N \in \mathbb{N}$ be the number of pixels in our images and $J := \{1, \dots, N\}$ the corresponding index set. Let $f \in \mathbb{R}^N$ be the

Table 5.1: Possible choices for tonal weights Ψ.

$\Psi(s^2)$		$\Psi'(s^2)$	known in the context of
s^2		1	Tikhonov regularisation [184]
$2\left(\sqrt{s^2+\lambda^2}-\lambda\right)$		$(s^2+\lambda^2)^{-\frac{1}{2}}$	regularised total variation [158, 1]
$2\lambda^2\left(\sqrt{1+\frac{s^2}{\lambda^2}}-1\right)$		$\left(1+\frac{s^2}{\lambda^2}\right)^{-\frac{1}{2}}$	nonlinear regularisation, Charbonnier et al. [39]
$\lambda^2\log\left(1+\frac{s^2}{\lambda^2}\right)$		$\left(1+\frac{s^2}{\lambda^2}\right)^{-1}$	nonlinear diffusion, Perona and Malik [154]
$2\lambda^2\left(1-\exp\left(-\frac{s^2}{2\lambda^2}\right)\right)$		$\exp\left(-\frac{s^2}{2\lambda^2}\right)$	nonlinear diffusion, Perona and Malik [154]
$\min(s^2,\lambda^2)$		$\begin{cases} 1 & \lvert s\rvert<\lambda \\ 0 & \text{else} \end{cases}$	segmentation, Mumford and Shah [142]

given image. In the following, $u \in \mathbb{R}^N$ stands for a filtered version of f and for the variable of the energy function E. The energy function E of the NDS filter presented in [139] can be decomposed into two parts: the *data* and the *smoothness term*.

The data term can be written as

$$E_D(u) \;=\; \sum_{i,j\in J} \Psi_D\left(|u_i - f_j|^2\right) w_D\left(|x_i - x_j|^2\right) \tag{5.1}$$

where $\Psi_D : [0,\infty) \longrightarrow [0,\infty)$ is an increasing function which plays the role of a penaliser for the difference between u and the initial image f, the so-called *tonal weight function*. This weight function is important for the behaviour of the filtering method with respect to image edges and the robustness against outliers. In principle we can use here all the penalising functions we already know from regularisation approaches. Six possibilities for such penalising functions are shown in Table 5.1. They are essentially the same as the ones presented in the first chapter. Here we have introduced some additional factors or constants in order to assure that the derivatives are exactly the same as some commonly used diffusivities. We notice that the data term not only compares the grey values of u and f at the pixel x_i, but it also takes a nonlocal neighbourhood into account. This neighbourhood is defined by the *spatial weight function* $w_D : [0,\infty) \longrightarrow [0,\infty)$ depending on the Euclidean distance between the pixels x_i and x_j. We have seen already in Table 1.4 how these functions can be chosen. By choosing special cases and parameters of w_D one can determine the amount of locality of the filter from one pixel up to the whole image domain.

The second ingredient of the NDS function is the smoothness term

$$E_S(u) \;=\; \sum_{i,j\in J} \Psi_S\left(|u_i - u_j|^2\right) w_S\left(|x_i - x_j|^2\right) \tag{5.2}$$

which differs from the data term by the fact that not the difference between u and f is calculated as argument of the penaliser, but the difference between grey values of u inside a larger neighbourhood. As penalising function Ψ_S and as spatial window w_S one can choose similar functions as described for Ψ_D and w_D, respectively. Since E_S does not depend on the initial image f, each constant image $u \equiv c \in \mathbb{R}$ would always yield a minimum: Thus we are rather interested in intermediate solutions during the minimisation process.

The complete NDS energy function is then the convex combination of these two parts:

$$\begin{aligned} E(u) \;&=\; \alpha\, E_D(u) \,+\, (1-\alpha)\, E_S(u) && (5.3)\\ &=\; \alpha \sum_{i,j\in J} \Psi_D\left(|u_i - f_j|^2\right) w_D\left(|x_i - x_j|^2\right) \\ &\quad + (1-\alpha) \sum_{i,j\in J} \Psi_S\left(|u_i - u_j|^2\right) w_S\left(|x_i - x_j|^2\right) && (5.4) \end{aligned}$$

for $0 \le \alpha \le 1$. This function combines two ways of smoothing: Besides the smoothness term, also choosing a window with larger size in the data term introduces some averaging over this neighbourhood. The presence of the initial image f in the data term makes nonflat minima of E possible.

5.1.2 Included Classical Methods

In the following, we give a short list of special methods included in the NDS framework:

- **Local M-smoothers / W-estimators:** Restricting the neighbourhood size leads to the data term

$$E_D(u) \;=\; \sum_{i,j\in J} \Psi_D(|u_i - f_j|^2) w_D(|x_i - x_j|^2) \tag{5.5}$$

  as shown in (5.1) where nonflat minimisers are possible. To search for a minimiser, we consider critical points with $\nabla E(u) = 0$ which is equivalent to

$$\sum_{j\in J} \Psi'_D(|u_i - f_j|^2) w_D(|x_i - x_j|^2)(u_i - f_j) \;=\; 0 \tag{5.6}$$

  for all $i \in J$. This can be understood as a fixed point equation $u = F(u)$ if we bring the vector u to the other side. The corresponding iterative scheme

is

$$u_i^{k+1} \;=\; \frac{\sum_{j\in J} \Psi'_D(|u_i^k - f_j|^2) w_D(|x_i - x_j|^2) f_j}{\sum_{j\in J} \Psi'_D(|u_i^k - f_j|^2) w_D(|x_i - x_j|^2)} \tag{5.7}$$

which has been considered as *local M-smoother* [41] or as localised version of the *W-estimator* [97, 210]. Doing only the first iteration leads to the classical sigma filter [114] as shown in (1.37).

- **Bilateral filter:** Performing the same steps as above with a smoothness term (5.2) as starting point, one can obtain the averaging filter

$$u_i^{k+1} \;=\; \frac{\sum_{j\in J} \Psi'_S(|u_i^k - u_j^k|^2) w_S(|x_i - x_j|^2) u_j^k}{\sum_{j\in J} \Psi'_S(|u_i^k - u_j^k|^2) w_S(|x_i - x_j|^2)} \;. \tag{5.8}$$

  This is an iterative version of the *bilateral filter* [185], *SUSAN filter* [169], or *nonlinear Gaussian filter* [8] already described in the first chapter. Since we start with a smoothness term, the initial image has to be taken as starting vector for the iterative scheme, and we are interested not in the steady state, but in intermediate solutions.

- **Bayesian / regularisation approaches:** Classical regularisation approaches can be expressed with the NDS function by using very small local neighbourhoods. Typically, one will use only the central pixel in the data term, and only the four direct neighbours $\mathcal{N}(i)$ of the pixel at position x_i in the smoothness term. This results in the energy function

$$E(u) \;=\; \alpha \sum_{i\in J} \Psi_D(|u_i - f_i|^2) + (1-\alpha) \sum_{i\in J} \sum_{j\in\mathcal{N}(i)} \Psi_S(|u_j - u_i|^2) \tag{5.9}$$

  which is some kind of discrete anisotropic regularisation function since the differences to all four neighbours are penalised independently. Exchanging the sum and the penaliser in the smoothness term would yield the isotropic variant with an approximation of $|\nabla u|^2$ inside the penaliser.

We see that the NDS functional allows to incorporate many simplification techniques in one model.

5.1.3 Parameter Selection

As we have seen in the last section, the NDS model is a very general model which is capable of yielding various kinds of filtering results depending on the choice of parameters. To obtain denoising results with good visual quality, one has to determine an appropriate set of parameters and weights depending on the noise and the properties of the desired result. For example, nonconvex tonal weights tend to yield images which can be seen as compositions of regions with constant

Figure 5.1: Example of the trade-off between different parameters. *Left:* Original image, 256×256 pixels, with additive Gaussian noise with standard deviation $\sigma = 50$. *Middle:* Denoising with E_D, radius $r_D = 5$. *Right:* Denoising with E_C, $\alpha = 0.424$.

grey value, while quadratic weights usually blur the image edges. In this context the question arises if there is some redundancy in the set of parameters. We will display one experiment concerning this question here – further experiments and remarks addressing the problem of choosing the parameters can be found in [155]. In our experiment we compare the influence of the neighbourhood size in the data term with the weight α between data and smoothness term. To this end we consider a pure data term

$$E_D(u) \;=\; \sum_{i,j\in J} (u_i - f_j)^2 w_D(|x_i - x_j|^2) \tag{5.10}$$

with a quadratic tonal penaliser and a disc-shaped hard window function w_D with radius r_D. On the other hand, we have a function with a local data term, and a smoothness term that involves only the direct neighbours:

$$E_C(u) \;=\; \alpha \sum_{i\in J} (u_i - f_i)^2 + (1-\alpha) \sum_{i\in J} \sum_{j\in\mathcal{N}(i)} (u_i - u_j)^2 \tag{5.11}$$

The only parameter α determines how smooth the result is in this case.

We are looking for a quantification of the difference between the results obtained by minimising these two functions. The connection between the parameters α and r_D with most similar results is especially interesting. Figure 5.1 shows that the filtering results obtained by the two approaches are hardly distinguishable when the parameters are suitably chosen. Figure 5.2 shows the graph relating the size of the window r_D and the value for α such that the difference is minimised.

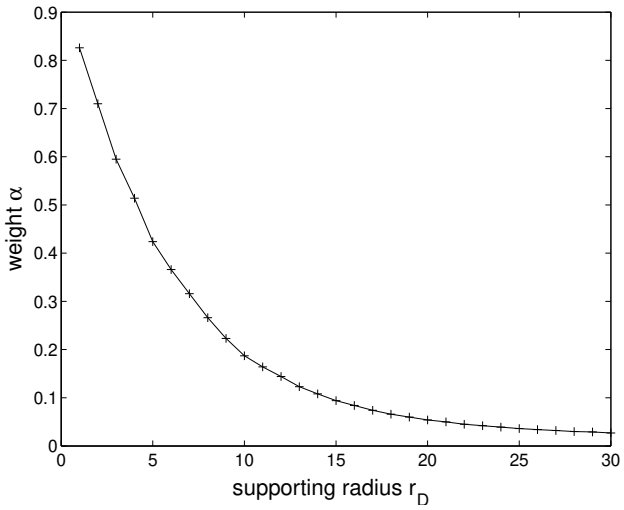


Figure 5.2: Trade-off between large kernels and small α. Optimal value of α depending on the radial support r_D.

5.2 Minimisation Methods

After discussing the derivation and the meaning of the NDS functional we now study different methods to minimise it. All numerical minimisation methods shown here based on conditions on the derivatives of E, so we now calculate the first and second partial derivatives of E.

Taking the partial derivatives of the data term (5.1) yields

$$\frac{\partial E_D}{\partial u_k} = 2\sum_{j\in J} \Psi'_D\left(|u_k - f_j|^2\right)(u_k - f_j) w_D\left(|x_k - x_j|^2\right) \tag{5.12}$$

$$\frac{\partial^2 E_D}{\partial u_k \partial u_r} = \begin{cases} 2\sum_{j\in J} \left[2\Psi''_D\left(|u_r - f_j|^2\right)(u_r - f_j)^2 \right. \\ \qquad \left. + \Psi'_D\left(|u_r - f_j|^2\right)\right] w_D\left(|x_r - x_j|^2\right) & r = k \\ 0 & r \neq k \end{cases} \tag{5.13}$$

In a similar way we calculate the derivatives of the smoothness term (5.2) which leads to

$$\frac{\partial E_S}{\partial u_k} = 4\sum_{j\in J} \Psi'_S\left(|u_k - u_j|^2\right)(u_k - u_j) w_S\left(|x_k - x_j|^2\right) \tag{5.14}$$

$$\frac{\partial^2 E_S}{\partial u_k \partial u_r} = \begin{cases} 4\sum_{j\in J}\left[2\Psi_S''\left(|u_r-u_j|^2\right)(u_r-u_j)^2 \right. \\ \qquad \left. +(1-\delta_{rj})\Psi_S'\left(|u_r-u_j|^2\right)\right] w_s\left(|x_r-x_j|^2\right) & r=k \\ -4\left[2\Psi_S''\left(|u_k-u_r|^2\right)(u_k-u_r)^2 \right. \\ \qquad \left. +\Psi_S'(|u_k-u_r|^2)\right] w_S\left(|x_k-x_r|^2\right) & r\neq k \end{cases} \tag{5.15}$$

In the second derivatives δ_{rj} denotes the Kronecker symbol $\delta_{rj} = \begin{cases} 1 & r=j \\ 0 & \text{else} \end{cases}$.

It is clear that the complete derivatives then have the form

$$\frac{\partial E}{\partial u_i} = \alpha\frac{\partial E_D}{\partial u_i} + (1-\alpha)\frac{\partial E_S}{\partial u_i} \ , \tag{5.16}$$

and the corresponding sum for the second derivatives, respectively. Having these derivatives at hand we can now study the concrete minimisation algorithms.

5.2.1 Jacobi Method

For a critical point u of the energy functional E we have

$$\nabla E(u) \ = \ 0 \quad \Longleftrightarrow \quad \frac{\partial E}{\partial u_i} \ = \ 0 \quad \text{for all } i\in J \ . \tag{5.17}$$

We define the abbreviations

$$d_{i,j} \ := \ \Psi_D'\left(|u_i-f_j|^2\right) w_D\left(|x_i-x_j|^2\right) \ , \tag{5.18}$$

$$s_{i,j} \ := \ \Psi_S'\left(|u_i-u_j|^2\right) w_S\left(|x_i-x_j|^2\right) \tag{5.19}$$

which help us to rewrite (5.17) as

$$0 \ = \ \alpha\sum_{j\in J} d_{i,j}(u_i-f_j) + 2(1-\alpha)\sum_{j\in J} s_{i,j}(u_i-u_j) \tag{5.20}$$

where we use the partial derivatives shown in (5.12) and (5.14). This can be transformed into fixed point form

$$u_i \ = \ \frac{\alpha\sum_{j\in J} d_{i,j}\, f_j + 2(1-\alpha)\sum_{j\in J} s_{i,j}\, u_j}{\alpha\sum_{j\in J} d_{i,j} + 2(1-\alpha)\sum_{j\in J} s_{i,j}} \ . \tag{5.21}$$

To have a positive denominator we assume that $\Psi_{\{D,S\}}'(s^2) > 0$, i. e., the penalisers are monotonically increasing. Furthermore we assume that $w_{\{D,S\}}(s^2) \geq 0$ and $w_{\{D,S\}}(0) > 0$ for the spatial weights. We use this equation to build up a first iterative method to minimise the value of E where the upper index k denotes the iteration number. Note that $d_{i,j}$ and $s_{i,j}$ also depend on the evolving image

u^k and thus also get a superscript to denote the iteration level involved. The corresponding fixed point iteration then reads as

$$u_i^0 := f_i \ , \tag{5.22}$$

$$u_i^{k+1} := \frac{\alpha \sum_{j\in J} d_{i,j}^k \, f_j + 2(1-\alpha) \sum_{j\in J} s_{i,j}^k \, u_j^k}{\alpha \sum_{j\in J} d_{i,j}^k + 2(1-\alpha) \sum_{j\in J} s_{i,j}^k} \ . \tag{5.23}$$

In the following we will write this scheme (5.23) in the form $u^{k+1} = F(u^k)$ with $F : \mathbb{R}^N \longrightarrow \mathbb{R}^N$. We note that in (5.23) we calculate u^{k+1} using only components of the vector u^k of the old iteration level:

$$u_i^{k+1} := F^i(u^k) \quad \text{for all } i \in J, k \in \mathbb{N} \ . \tag{5.24}$$

Such a method can also be called a **nonlinear Jacobi method**.

Proposition 5.1 (Maximum-Minimum Principle)
With the assumptions on $\Psi_{\{D,S\}}$ and $w_{\{D,S\}}$ from above, the scheme (5.23) staisfies a maximum-minimum principle.

Proof: With our assumptions on $\Psi_{\{D,S\}}$ and $w_{\{D,S\}}$ from above we know that $d_{i,j}^k \geq 0$ and $s_{i,j}^k \geq 0$ for all i, j, k. That means in (5.23), u_i^{k+1} is calculated as a convex combination of grey values of the initial image f and of the last iteration step u^k. Thus we have

$$\min_{j\in J}\{u_j^k, f_j\} \ \leq \ u_i^{k+1} \ \leq \ \max_{j\in J}\{u_j^k, f_j\} \quad \text{for all } i \in J, k \in \mathbb{N} \ . \tag{5.25}$$

Induction shows that the fixed point scheme (5.23) satisfies a maximum-minimum principle, i. e.

$$\min_{j\in J}\{f_j\} \ \leq \ u_i^k \ \leq \ \max_{j\in J}\{f_j\} \quad \text{for all } i \in J, k \in \mathbb{N} \ . \tag{5.26}$$

□

In the next proposition, we see that this property is not only useful from a practical point of view: Together with continuity, it gives us the existence of a fixed point.

Proposition 5.2 (Existence of a Fixed Point)
The fixed point equation (5.21) has a solution.

Proof: Let us consider the set $M := \left\{u \in \mathbb{R}^N \mid \|u\|_\infty \leq \|f\|_\infty\right\}$ with the norm $\|u\|_\infty := \max_{j\in J} |u_j|$. M is nonempty, compact and convex. Then the maximum-minimum stability implies that $F(M) \subseteq M$. With our requirements on $\Psi_{\{D,S\}}$ and $w_{\{D,S\}}$, the denominator in (5.23) is always larger than zero. This means that each component $F_i : \mathbb{R}^N \longrightarrow \mathbb{R}$ is continuous with respect to the norm

$\|\cdot\|_\infty$. Since this holds for all i, we know that $F : (\mathbb{R}^N, \|\cdot\|_\infty) \longrightarrow (\mathbb{R}^N, \|\cdot\|_\infty)$ is continuous. Then Brouwer's fixed point theorem (see [15] or [214, page 51], for example) shows that F has a fixed point in M. □

From the derivation it is clear that a fixed point corresponds to a critical point of E. If we have chosen our penaliser functions such that the energy function is strictly convex, this is equivalent to the unique minimum of E.

5.2.2 Gauß-Seidel Method

Instead of the nonlinear Jacobi method (5.24) one can also use a nonlinear Gauß-Seidel method which involves pixels of the old and the new iteration level. For each pixel $u_i =: x^0$, we perform m steps of a local fixed point iteration

$$x^{r+1} := F^i(u_1^{k+1}, \ldots, u_{i-1}^{k+1}, x^r, u_{i+1}^k, \ldots, u_N^k) \quad r = 1, \ldots, m-1 \tag{5.27}$$

and set $u_i^{k+1} := x^m$ afterwards. Since these inner steps satisfy a maximum-minimum principle, the whole Gauß-Seidel method does. Thus one can apply the same reasoning as above and gets the existence of fixed points for the equation.

5.2.3 Newton's Method

We search a zero of the gradient $\nabla E(u) = 0$. To this end we use Newton's method for the function ∇E:

$$u^{k+1} \;=\; u^k - H(E, u^k)^{-1}\, \nabla E(u^k) \;, \tag{5.28}$$

where $H(E, u^k)$ is the Hessian of E at the point u^k. In each step of (5.28) we have to solve a linear system of equations. This system of equations can only be solved if the Hessian is invertible which is the case for a strictly convex function E. That means we can not use Newton's method for all penalisers shown in the last section. If both $\Psi_D(s^2)$ and $\Psi_S(s^2)$ are strictly convex in s, i. e. $2\Psi''(s^2)s^2 + \Psi'(s^2) > 0$, the Hessian $H(E, u^k)$ has positive diagonal entries and is strictly diagonally dominant. This does not only allow us to solve the linear system of equations, but it also gives us the possibility to use a whole variety of iterative solution algorithms like the Gauß-Seidel, successive overrelaxation, or conjugate gradient method (see [177, 160], for example). We have chosen to use the Gauß-Seidel method here to solve the linear system of equations since it does not introduce further numerical parameters besides the number of iterations.

A practical observation shows that the steps of Newton's method are often too long. Thus we have used a simple line-search strategy:

$$u^{k+1} \;=\; u^k - \sigma_k H(E, u^k)^{-1}\, \nabla E(u^k) \tag{5.29}$$

with $\sigma_k \in (0, 1]$. We try $\sigma_k = 1, \frac{1}{2}, \frac{1}{4}, \ldots$ until the energy is decreasing in the step: $E(u^{k+1}) < E(u^k)$.

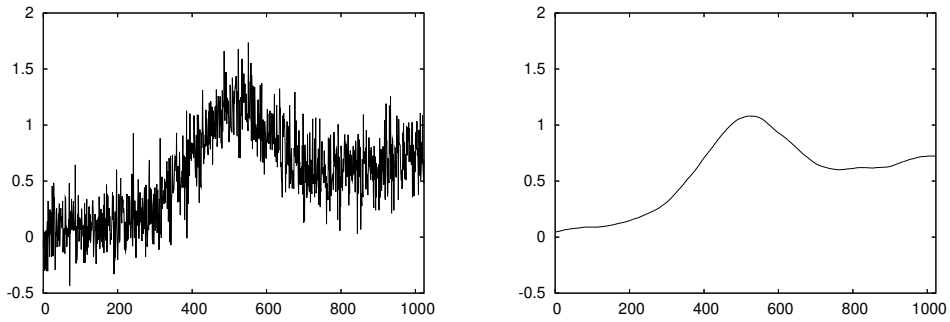


Figure 5.3: Denoising experiment in 1-D. Left: Test signal with additive Gaussian noise with zero mean, size 1024 pixels. Right: Denoised version of the signal.

It is clear that one step of Newton's method is much more expensive than one fixed point iteration step. Nevertheless, numerical examples will show that the whole process can still converge faster.

5.2.4 Gauß-Seidel Newton Method

Here we solve the single component equations with Newton's method. We start with the pixel value $x^0 = u_i^k$ of the last iteration level and set

$$x^{r+1} = x^r - \sigma_l \left(\frac{\partial^2 E}{\partial u_i^2}(\tilde{u}) \right)^{-1} \frac{\partial E}{\partial u_i}(\tilde{u}) \tag{5.30}$$

with $\tilde{u} = (u_1^{k+1}, \ldots, u_{i-1}^{k+1}, x^r, u_{i+1}^k, \ldots, u_N^k)$. After m steps of this method we set $u_i^{k+1} = x^m$ and proceed with the next pixel. The only difference is that we use the criterion $E_{loc}(x^{r+1}) < E_{loc}(x^r)$ for the choice of the step size σ_r where the local energy is defined as

$$\begin{aligned} E_{loc}(u) &= \alpha \sum_{j \in J} \Psi_D \left(|x^r - f_j|^2 \right) w_D \left(|x_i - x_j|^2 \right) \\ &\quad + (1-\alpha) \sum_{j \in J} \Psi_S \left(|x^r - \tilde{u}_j|^2 \right) w_S \left(|x_i - x_j|^2 \right) \quad . \end{aligned} \tag{5.31}$$

We should note that besides the number of (outer) iterations, all methods except of the Jacobi method have the number of inner iterations as an additional parameter for the numerics.

5.3 Numerical Experiments

Now we investigate the practical behaviour of the methods presented in the last section. We use the two stopping criteria

$$\|u^{k+1} - u^k\|_2 < a \quad \text{and} \quad |E(u^{k+1}) - E(u^k)| < b \ .$$

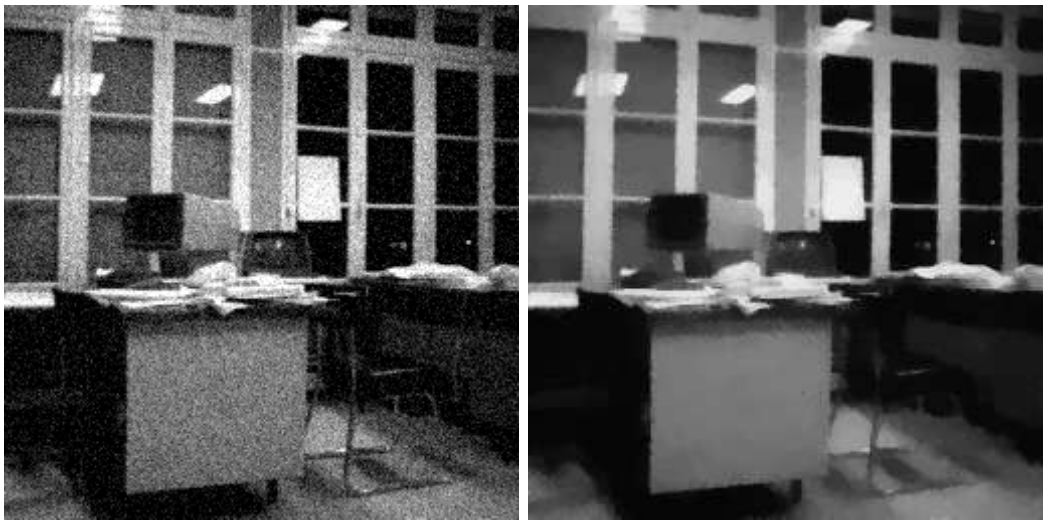


Figure 5.4: Denoising experiment in 2-D with Gaussian noise. Left: Test image (256 × 256 pixels) with additive Gaussian noise of standard deviation $\sigma = 20$. Right: Denoised version of the image.

That means we stop the algorithm if the changes of both the evolving image (in terms of the Euclidean norm) and the energy value are smaller than prescribed limits a and b. The results of the 1-D example are displayed in Figure 5.3 and Table 5.2. Here we have Gaussian noise, and we have chosen $\Psi_D(s^2) = s^2$, $\Psi_S(s^2) = 2\left(\sqrt{s^2 + \lambda^2} - \lambda\right)$ with $\lambda = 0.01$, and $w_D(s^2) = w_S(s^2) = 1$ inside a data term window of size 7 and a smoothness term window of size 11 with $\alpha = 0.5$. The number of inner iterations was optimised to yield a fast convergence for each method. We see that Newton's method is the fastest one in this case while all of the methods yield almost equal ℓ^1-errors.

We see that results of our first 2-D experiment in Figure 5.4 and Table 5.3. Since we also have Gaussian noise here, we also use a quadratic data term and a regularised total variation penaliser in the smoothnes term as in the previous 1-D experiment. The regularisation parameter for the smoothness term penaliser is $\lambda = 0.1$ here. The data term is local in this case, and the smoothness term has

Table 5.2: Denoising experiment in 1-D with $a = 10^{-2}$ and $b = 10^{-6}$.

method	iterations	inner it.	energy	ℓ^1-error	time [sec]
original signal	–	–	838.21660	163.994	–
Jacobi	1309	–	165.70820	23.167	3.301
Newton	25	60	165.70807	23.252	0.510
Gauß-Seidel	842	1	165.70815	23.200	2.354
G.-S. Newton	683	1	165.70813	23.229	5.739

Table 5.3: Denoising experiment in 2-D with Gaussian noise and $a = b = 500$.

method	iterations	inner it.	energy	ℓ^1-error	time [sec]
original image	–	–	$4.194 \cdot 10^7$	90008.1	–
Jacobi	292	–	$2.525 \cdot 10^7$	54304.9	77.189
Newton	31	5	$2.542 \cdot 10^7$	51466.2	47.223
Gauß-Seidel	48	10	$2.519 \cdot 10^7$	57677.9	73.335
G.-S. Newton	70	1	$2.690 \cdot 10^7$	54296.9	287.133

Figure 5.5: Denoising experiment in 2-D with impulse noise. Left: Test image (256×256 pixels) with salt-and-pepper noise. Right: Denoised version of the image.

Table 5.4: Denoising experiment in 2-D with impulse noise and $a = b = 100$.

method	iterations	inner it.	energy	ℓ^1-error	time [sec]
original image	–	–	$2.753 \cdot 10^7$	61232.9	–
Jacobi	47	–	$1.901 \cdot 10^7$	17964.3	9.32
Newton	409	2	$1.917 \cdot 10^7$	18351.6	938.95
Gauß-Seidel	8	10	$1.901 \cdot 10^7$	18033.8	8.75
G.-S. Newton	3	10	$1.901 \cdot 10^7$	18047.7	76.77

size 5. The regularisation weight is $\alpha = 0.5$. We see that also in 2-D, Newton's method converges faster than the others.

This is not the case for all configurations: Figure 5.5 and Table 5.4 contain the results of the second experiment in 2-D. For the removal of salt-and-pepper noise we chose $\Psi_D(s^2) = 2\left(\sqrt{s^2 + \lambda^2} - \lambda\right)$ with $\lambda = 0.01$, $\Psi_S(s^2) = \lambda^2\left(\sqrt{1 + \frac{s^2}{\lambda^2}} - 1\right)$ with $\lambda = 0.1$. We set $w_D(s^2) = w_S(s^2) = 1.0$ with both windows of size 3 and $\alpha = 0.95$. As the data term contains the regularised total variation penaliser, this filter is related to a median filter with small smoothness term contribution. Here we observe the opposite case than in the previous examples: The simple Gauß-Seidel and Jacobi iterations are faster than Newton's method.

5.4 Summary

In this chapter, we have investigated four different algorithmic approaches for the image simplification NDS-model presented in [139]. We have given an overview over the spectrum of methods that can be represented with this model and performed experiments concerning the selection of parameters. For schemes based on a nonlinear Jacobi method we have shown the existence of fixed points. Newton's method is only applicable for a certain class of convex penalisers. We have seen with practical examples that in terms of running time we can not prefer one single method in general. It is an interesting question if coarse-to-fine strategies could be applied in this context to further speed up the minimisation.

Chapter 6

Conclusions and Outlook

In this chapter, we conclude the thesis by giving a short summary of the main results connected with a glimpse at some open questions arising in the respective context.

6.1 Conclusions

This thesis has dealt with several types of methods for image simplification, smoothing and denoising. After introducing classical image processing methods in Chapter 1, we have discussed derivative-based filtering approaches in Chapter 2. Firstly we have investigated linear scale-spaces arising from fractional order regularisation and diffusion filters in Section 2.1. Special emphasis was on the fact that we have considered combinations of several fractional orders here. We have seen that such combinations in general contradict scale invariance. Nevertheless, the experiments in Subsection 2.1.5 have shown that higher order filters need not to violate a maximum-minimum principle: It is possible to construct linear combinations of different derivative orders which yield positive convolution kernels and thus satisfy a maximum-minimum principle.

In the nonlinear case in Section 2.2, we have investigated nonquadratic regularisation and nonlinear evolution equations with penalisers and diffusivities depending on all squared partial derivatives of a certain integer order. We have seen that these filters are stable in the continuous or discrete 2-norm and preserve higher moments of the data. To find the higher order analogue to the edge enhancement of classical diffusion equations, a consideration of the behaviour in terms of forward and backward diffusion has shown to be helpful. Experimentally, nonlinear diffusion filtering of order $2p$ can leads to piecewise polynomial results of degree $p-1$. An important example is the fourth-order diffusion equation which allows for the recovery of images consisting in linear regions. Experiments show that it is possible with such a filter and the Perona-Malik diffusivity to denoise an image in an edge-preserving manner without staircasing artefacts.

Besides this regularisation and generalised diffusion approaches, we have discussed variational approaches involving higher order data terms. We have reviewed two ways of implementations for the corresponding filters. In case of the total variation penaliser, they have shown to produce significantly less blurring of image edges.

In Chapter 3 we have discussed the connection between multiscale wavelet shrinkage and integrodifferential equations. The key for the understanding of these connections has been to see the wavelets as derivatives of smoothing kernels. This allows to consider the wavelet transform as presmoothed derivative operator. The backtransform additionally integrates over all scales turning the nonlinear diffusion equation into an integrodifferential equation. A similar formulation than in the continuous setting has also been used in the discrete case. Here, we have focused on the shape of the convolution kernels appearing in the integrodifferential equations since there is no exact dilation operation, but only an approximative one. Besides the 1-D case, we have also taken a look at two-dimensional filtering methods. Numerical experiments have shown that only using larger scales makes it impossible to remove high-frequent components of the noise. Involving multiple dyadic scales helps to speed up the denoising process and can even help to enhance the quality in some examples.

Adaptive averaging filters and some related PDE models have been the subject of Chapter 4. We have seen that a simple 1-D adaptive averaging method can be related to what we have called accelerated Perona-Malik diffusion. The additional factor on the right-hand side leads to increased edge enhancement. Nevertheless, there are several possibilities to transfer this idea to 2-D which have been sketched here. We have seen that a filter closely connected to the integration model for anisotropic diffusion can be obtained as scaling limit for the bilateral filter. Motivated by the 1-D scaling limit, a straight-forward generalisation to 2-D yields the generalised mean curvature motion. The GMCM filter family has shown to yield interesting results from a practical point of view: The filters behave similar to mean curvature motion, while the edges in the image remain sharp. By using a diffusivity function with a contrast parameter like the Perona-Malik diffusivity, one can select objects in the image to be shrunken in an MCM-like manner and others to be blurred very fast similar to linear diffusion.

Chapter 5 has given an introduction to the NDS filtering framework and its connections to classical image processing methods. We have focused on its numerical implementation and properties of this methods. For example, an approach based on a fixed point iteration or nonlinear Jacobi method satisfies the maximum-minimum property. Other approaches based on Newton's method can show faster convergence for several problem instances. We have also discussed the choice of parameters with the help of numerical experiments. This helps to understand the redundancy in the parameter set of NDS filtering.

In general, we have reviewed many different ways for filtering digital data while preserving or even enhancing its important features. We have tried to describe

the properties of these filters from a rather practical point of view together with possible ways of implementation. It is the hope that some of the ideas presented in this thesis can be useful for solving practical application problems in image processing or as an ingredient in computer vision systems.

6.2 Outlook

In the previous section, we have summarised the main contributions of this thesis. Let us now sketch some interesting questions for future research:

Concerning the research on linear scale-spaces, the fact that combinations of a high and a low filter order are observed to satisfy a maximum-minimum principle deserves a further investigation: To derive a theoretical background for this effect and possibly also deduce limits for the weights depending on the scale-space order is an open problem with practical relevance. This question is strongly connected with Bochner's theorem and the notion of positive functions which are functions whose Fourier transform is nonnegative [132, 13]. Unfortunately, this notion is usually used the other way round: Knowing that the Fourier transform is nonnegative allows to determine that a function is positive. To solve our question here, we would have to go in the other direction.

The connection between variational methods with higher order data terms and the classical approaches has been investigated only for ℓ^1 regularisation so far [59]. The connection of these methods to the ones described before still needs further research in a more general framework. Besides derivatives, one can also think of different of operators in the data term like convolution operators, for example.

In recent years, other kinds of data than grey and color images, for example matrix-valued data sets, become more and more important for practical purposes. Examples for matrix-valued data include diffusion tensor magnetic resonance imaging in medicine or stress tensors to measure deformations in material sciences. The higher order diffusion filters described in this thesis could be carried over to matrix-valued data in a similar way as proposed by Burgeth et al. [24, 23, 27] for the classical Perona-Malik filters. One of the most important questions in this context is how suitable stability notions look like and how corresponding criteria, for example for explicit discretisations, can be formulated in this framework. Faster implementations, for example semi-implicit schemes, are also not available so far.

In our research on the connections between wavelets and diffusion filters in Chapter 3, we have mainly worked in one dimension and only considered isotropic filters so far. Further considerations in the 2-D case and involving anisotropy would be interesting extensions. The natural candidate for such an extension on the side of PDEs would be anisotropic nonlinear diffusion as described in Subsection 1.2.3. On the wavelet side we have already mentioned several approaches

like curvelets or orientation scores in Section 1.3.

The description of the GMCM method in Chapter 4, we have shown that this filter class has interesing properties in practice. The exact shrinkge velocity is only one of the theoretical properties which are still open in this context. To deduce precise extinction times for objects of certain diameter would make these methods much more appealing for practical purposes.

Concerning the numerical minimisation of the NDS function in Chapter 5, there are also other approaches which deserve a closer investigation. An interesting question is the usefulness of multigrid methods in this context. At least for small windows, this procedure can be expected to speed up the convergence. Also for larger windows, coarse-to-fine strategies leading to better starting points on the finer scales could be worth a closer investigation.

Appendix A

Contributions and Publications

A.1 Further Contributions

The main contributions of this thesis have already been sketched in Chapter 6. Furthermore, as co-author of several publications, there have been some contributions to different topics during my studies which have not been included in detail in this thesis. In this section, a short sketch of these topics is given:

- **Linear scale-spaces:** With the so-called *relativistic scale-spaces*, a new family of linear scale-spaces has been described [26]. The integral kernel can be given in explicit form in this case.

  For linear scale-spaces, usually the semigroup property is satisfied by using kernels with Fourier transforms of exponential form. The *Bessel scale-space* is the first example for a linear scale-space satisfying the semigroup property with another structure of the kernels [25].

- **TV regularisation:** The connection between discrete higher order TV regularisation, support vector regression and discrete splines have been investigated [174, 175]. In fact, there are different equivalent formulations for the same application in terms of sparse representation, contact problems and support vector regression. For higher order data terms, one obtains discrete splines with a defect [59].

- **Applications of higher order methods:** Besides the problem of image denoising, higher order regularisation and diffusion can also be useful for other problems in image processing and computer vision. Concerning the optic flow problem, one can think of both higher order data and smoothness terms [151]. For the problem of shape from shading, a common model gives equations involving the first derivatives of the depth information as desired quantity. That means, the first derivative is already determined by the constraints of the problem. In this context, it is a natural consequence to use a higher order smoothness requirement [193].

- **Processing of matrix-valued data:** For the processing of matrix-valued data, one has often generalised PDE-based approaches in the same way as for colour images. Nevertheless, this does not reflect the understanding of matrices as operators and the meaning of the different matrix entries in this context. Recently, it has been proposed to understand the equations and all functions involved there already as matrix-valued. This offers a different approach for filtering and also for numerical treatment. Several filter classes such as PDE-based morphology, Perona-Malik filtering or coherence-enhancing anisotropic diffusion, have been transferred to this setting [22, 24, 23, 27].

A.2 Publications

Journal Articles:

- N. Papenberg, A. Bruhn, T. Brox, S. Didas, J. Weickert: Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, Vol. 67, No. 2, 141–158, April 2006. **Invited paper.**

- G. Steidl, S. Didas, J. Neumann: Splines in higher order TV regularization. *International Journal of Computer Vision*, Vol. 70, No. 3, 241–255, 2006. **Invited paper.**

- B. Burgeth, A. Bruhn, S. Didas, J. Weickert, M. Welk: Morphology for tensor data: Ordering versus PDE-based approach. *Image and Vision Computing*, Vol. 25, No. 4, 496–511, April 2007.

- S. Didas, J. Weickert: Integrodifferential Equations for Continuous Multiscale Wavelet Shrinkage. *Inverse Problems and Imaging*, Vol. 1, No. 1, 47–62, 2007.

- S. Didas, S. Setzer, G. Steidl: Combined ℓ^2 data and gradient fitting in conjunction with ℓ^1 regularization. *Advances in Computational Mathematics*, 2007, in print.

Conference Papers:

- S. Didas, B. Burgeth, A. Imiya, J. Weickert: Regularity and scale-space properties of fractional high order linear filtering. In R. Kimmel, N. Sochen, J. Weickert (Eds.): *Scale-Space and PDE Methods in Computer Vision*. Lecture Notes in Computer Science, Vol. 3459, Springer, Berlin, 13–25, 2005.

- B. Burgeth, S. Didas, J. Weickert: Relativistic scale-spaces. In R. Kimmel, N. Sochen, J. Weickert (Eds.): *Scale-Space and PDE Methods in Computer Vision*. Lecture Notes in Computer Science, Vol. 3459, Springer, Berlin, 1–12, 2005.

- G. Steidl, S. Didas, J. Neumann: Relations between higher order TV regularization and support vector regression. In R. Kimmel, N. Sochen, J. Weickert (Eds.): *Scale-Space and PDE Methods in Computer Vision*. Lecture Notes in Computer Science, Vol. 3459, Springer, Berlin, 515–527, 2005.

- S. Didas, J. Weickert, B. Burgeth: Stability and local feature enhancement of higher order nonlinear diffusion filtering. In W. Kropatsch, R. Sablatnig, A. Hanbury (Eds.): *Pattern Recognition*. Lecture Notes in Computer Science, Vol. 3663, Springer, Berlin, 251–258, 2005.

- B. Burgeth, S. Didas, J. Weickert: The Bessel scale-space. In O. F. Olsen, L. Florack, A. Kuijper (Eds.): *Deep Structure, Singularities, and Computer Vision*. Lecture Notes in Computer Science, Vol. 3753, Springer, Berlin, 84–95, 2005.

- S. Didas, P. Mrázek, J. Weickert: Energy-based image simplification with nonlocal data and smoothness terms. In A. Iske, J. Levesley (Eds.): *Algorithms for Approximation*, 50–60, Springer, Heidelberg, 2006.

- S. Didas, J. Weickert: From adaptive averaging to accelerated nonlinear diffusion filtering. In K. Franke, K.-R. Müller, B. Nickolay, R. Schäfer (Eds.): *Pattern Recognition*. Lecture Notes in Computer Science, Vol. 4174, 101–110, Springer, Berlin, 2006.

- L. Pizarro, S. Didas, F. Bauer, J. Weickert: Evaluating a general class of filters for image denoising. In B. K. Ersbøll, K. S. Pedersen, S. I. Olsen (Eds.): *Image Analysis*, Lecture Notes in Computer Science, Vol. 4522, 601–610, Springer, Berlin, 2007.

- B. Burgeth, S. Didas, L. Florack, J. Weickert: A generic approach for singular PDEs for the processing of matrix fields. In A. Murli, N. Paragios, F. Sgallari (Eds.): *Scale Space and Variational Methods in Computer Vision*. Lecture Notes in Computer Science, Vol. 4485, 556–567, Springer, Berlin, 2007.

- S. Didas, J. Weickert: Combining curvature motion and edge-preserving denoising. In A. Murli, N. Paragios, F. Sgallari (Eds.): *Scale Space and Variational Methods in Computer Vision*. Lecture Notes in Computer Science, Vol. 4485, 568–579, Springer, Berlin, 2007.

- O. Vogel, A. Bruhn, J. Weickert, S. Didas: Direct shape-from-shading with adaptive higher order regularisation. In A. Murli, N. Paragios, F. Sgallari (Eds.): *Scale Space and Variational Methods in Computer Vision*. Lecture Notes in Computer Science, Vol. 4485, 871–882, Springer, Berlin, 2007.

Technical Reports:

- B. Burgeth, S. Didas, L. Florack, J. Weickert: A generic approach to diffusion filtering of matrix-fields. Technical Report No. 191, Department of Mathematics, Saarland University, Saarbrücken, Germany, March 2007.

- B. Burgeth, S. Didas, J. Weickert: A general structure tensor concept and coherence-enhancing diffusion filtering for matrix fields. Technical Report No. 197, Department of Mathematics, Saarland University, Saarbrücken, Germany, July 2007.

Theses:

- S. Didas: Higher order variational methods for noise removal in signals and images. Diplomarbeit, Fachrichtung Mathematik, Universität des Saarlandes, Saarbrücken, Germany, April 2004.

- S. Didas: Synchronisation in the Network-Multimedia Middleware (NMM). Fortgeschrittenenpraktikum, Fachrichtung Informatik, Universität des Saarlandes, Saarbrücken, Germany, Oktober 2002. Accepted as Bachelor's thesis, Oktober 2004.

Bibliography

[1] R. Acar and C. R. Vogel. Analysis of bounded variation penalty methods for ill-posed problems. *Inverse Problems*, 10:1217–1229, 1994.

[2] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel. Axioms and fundamental equations of image processing. *Archive for Rational Mechanics and Analysis*, 123:199–257, 1993.

[3] L. Alvarez, P.-L. Lions, and J.-M. Morel. Image selective smoothing and edge detection by nonlinear diffusion II. *SIAM Journal on Numerical Analysis*, 29(3):845–866, June 1992.

[4] F. Andreu, C. Ballester, V. Caselles, and J. M. Mazón. Minimizing total variation flow. *Differential and Integral Equations*, 14(3):321–360, March 2001.

[5] A. M. Arthurs. *Calculus of Variations*. Routledge & Kegan Paul Ltd., 1975.

[6] R. Ashino, S. J. Desjardins, C. Heil, M. Nagase, and R. Vaillancourt. Pseudo-differential operators, microlocal analysis and image restoration. In R. Ashino, P. Boggiatto, and M.-W. Wong, editors, *Advances in Pseudo-Differential operators*, volume 155 of *Operator Theory: Advances and Applications*, pages 187–202. Birkhäuser, Basel, 2004.

[7] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing – Partial Differential Equations and the Calculus of Variations*. Springer, Berlin, 2002.

[8] V. Aurich and J. Weule. Non-linear Gaussian filters performing edge preservating diffusion. In G. Sagerer, S. Posch, and F. Kummert, editors, *Mustererkennung*, Informatik Aktuell, pages 538–545. Springer, 1995.

[9] J. Bai and X.-C. Feng. Fractional order anisotropic diffusion for image denoising. *IEEE Transactions on Image Processing*, 2007, to appear.

[10] Y. Bao and H. Krim. Towards bridging scale-space and multiscale frame analyses. In A. A. Petrosian and F. G. Meyer, editors, *Wavelets in Signal and Image Analysis*, volume 19 of *Computational Imaging and Vision*, chapter 6, pages 169–192. Kluwer, Dordrecht, 2001.

[11] D. Barash. A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):844–847, June 2002.

[12] M. Belloni and B. Kawohl. A direct uniqueness proof for equations involving the p-Laplace operator. *Manuscripta Mathematica*, 109(2):229–231, October 2002.

[13] S. Bochner. *Lectures on Fourier Integrals*. Princeton University Press, New Jersey, 1959.

[14] K. Bredies, D. A. Lorenz, and P. Maass. Mathematical concepts of multiscale smoothing. *Applied and Computational Harmonic Analysis*, 19(2):141–161, August 2005.

[15] L. E. J. Brouwer. Über Abbildungen von Mannigfaltigkeiten. *Mathematische Annalen*, 71(1):97–115, March 1911.

[16] T. Brox and D. Cremers. Iterated nonlocal means for texture restoration. In F. Sgallari, A. Murli, and N. Paragios, editors, *Scale Space and Variational Methods in Computer Vision*, volume 4485 of *Lecture Notes in Computer Science*, pages 13–24. Springer, Berlin, 2007.

[17] T. Brox, R. van de Boomgaard, F. Lauze, J. van de Weijer, J. Weickert, P. Mrázek, and P. Kornprobst. Adaptive structure tensors and their applications. In J. Weickert and H. Hagen, editors, *Visualization and Processing of Tensor Fields*, Mathematics and Visualization, pages 17–47. Springer, Berlin, 2006.

[18] A. Buades, B. Coll, and J.-M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling and Simulation*, 4(2):490–530, 2005.

[19] A. Buades, B. Coll, and J.-M. Morel. Neighborhood filters and PDEs. *Numerische Mathematik*, 105(1):1–34, 2006.

[20] A. Buades, B. Coll, and J.-M. Morel. The staircasing effect in neighborhood filters and its solution. *IEEE Transactions on Image Processing*, 15(6):1499–1505, 2006.

[21] M. Burger, G. Gilboa, S. Osher, and J. Xu. Nonlinear inverse scale space methods. *Communications in Mathematical Sciences*, 4(1):179–212, 2006.

[22] B. Burgeth, A. Bruhn, S. Didas, J. Weickert, and M. Welk. Morphology for tensor data: ordering versus PDE-based approach. *Image and Vision Computing*, 25(4):496–511, April 2007.

[23] B. Burgeth, S. Didas, L. Florack, and J. Weickert. A generic approach to diffusion filtering of matrix fields. Technical Report 191, Department of Mathematics, Saarland University, Saarbrücken, Germany, March 2007.

[24] B. Burgeth, S. Didas, L. Florack, and J. Weickert. Singular PDEs for the processing of matrix-valued data. In F. Sgallari, A. Murli, and N. Paragios, editors, *Scale Space and Variational Methods in Computer Vision*, volume 4485 of *Lecture Notes in Computer Science*, pages 556–567. Springer, Berlin, 2007.

[25] B. Burgeth, S. Didas, and J. Weickert. The Bessel scale-space. In O. F. Olsen, L. Florack, and A. Kuijper, editors, *Deep Structure, Singularities, and Computer Vision*, volume 3753 of *Lecture Notes in Computer Science*, pages 84–95. Springer, Berlin, 2005.

[26] B. Burgeth, S. Didas, and J. Weickert. Relativistic scale-spaces. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*, pages 1–12. Springer, Berlin, 2005.

[27] B. Burgeth, S. Didas, and J. Weickert. A general structure tensor concept and coherence-enhancing diffusion filterin for matrix fields. Technical Report 197, Department of Mathematics, Saarland University, Saarbrücken, Germany, July 2007.

[28] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31(4):532–540, 1983.

[29] E. J. Candés and F. Guo. New multiscale transforms, minimum total variation synthesis: Applications to edge-preserving image reconstruction. *Signal Processing*, 82(11):1519–1543, 2002.

[30] R. A. Carmona and S. Zhong. Adaptive smoothing respecting feature directions. *IEEE Transactions on Image Processing*, 7(3):353–358, March 1998.

[31] V. Caselles, J.-M. Morel, and C. Sbert. An axiomatic approach to image interpolation. *IEEE Transactions on Image Processing*, 7(3):376–386, March 1998.

[32] F. Catté, P.-L. Lions, J.-M. Morel, and T. Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical Analysis*, 29(1):182–193, February 1992.

[33] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20:89–97, 2004.

[34] A. Chambolle, R. DeVore, N.-Y. Lee, and B. J. Lucier. Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transactions on Image Processing*, 7:319–335, 1998.

[35] A. Chambolle and B. L. Lucier. Interpreting translation-invariant wavelet shrinkage as a new image smoothing scale space. *IEEE Transactions on Image Processing*, 10:993–1000, 2001.

[36] T. Chan, S. Esedoglu, F. Park, and A. Yip. Total variation image restoration: overview and recent developments. In N. Paragios, Y. Chen, and O. Faugeras, editors, *Handbook of Mathematical Models in Computer Vision*, pages 17–32. Springer, New York, 2006.

[37] T. Chan, A. Marquina, and P. Mulet. High-order total variation-based image restoration. *SIAM Journal of Scientific Computing*, 22(2):503–516, 2000.

[38] T. F. Chan and H. M. Zhou. Total variation improved wavelet thresholding in image compression. In *Proc. Seventh International Conference on Image Processing*, volume II, pages 391–394, Vancouver, Canada, September 2000.

[39] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. *Proc. IEEE International Conference on Image Processing (ICIP-94, Austin, Nov. 13-16, 1994)*, 2:168–172, 1994.

[40] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, February 1997.

[41] C. K. Chu, I. K. Glad, F. Godtliebsen, and J. S. Marron. Edge-preserving smoothers for image processing. *Journal of the American Statistical Association*, 93(442):526–541, June 1998.

[42] A. Cohen, W. Dahmen, I. Daubechies, and R. DeVore. Harmonic analysis in the space BV. *Revista Matematica Iberoamericana*, 19:235–262, 2003.

[43] A. Cohen, R. DeVore, P. Petrushev, and H. Xu. Nonlinear approximation and the space $\mathrm{BV}(\mathrm{R}^2)$. *American Journal of Mathematics*, 121:587–628, 1999.

[44] R. R. Coifman and D. Donoho. Translation invariant denoising. In A. Antoine and G. Oppenheim, editors, *Wavelets in Statistics*, pages 125–150. Springer, New York, 1995.

[45] R. R. Coifman and A. Sowa. Combining the calculus of variations and wavelets for image enhancement. *Applied and Computational Harmonic Analysis*, 9(1):1–18, July 2000.

[46] R. R. Coifman and A. Sowa. New methods of controlled total variation reduction for digital functions. *SIAM Journal on Numerical Analysis*, 39(2):480–498, 2001.

[47] G. Cong, M. Esser, B. Parvin, and G. Bebis. Shape metamorphism using p-Laplacian equation. In *ICPR 2004. Proceedings of the 17th International Conference on Pattern Recognition*, volume 4, pages 15–18, 2004.

[48] R. E. Crochiere and L. R. Rabiner. Digital coding of speech in sub-bands. *Bell Systems Technical Journal*, 55(8):1069–1085, October 1976.

[49] A. Croisier, D. Esteban, and C. Galand. Perfect channel splitting by use of interpolation / decimation / tree decomposition techniques. In *International Conference on Information Sciences and Systems*, pages 443–446, Patras, Greece, August 1976.

[50] E. Cuesta and J. Finat. Image processing by means of a linear integro-differential equation. In M. M. Hamza, editor, *Visualization, Imaging and Image Processing*, pages 438–442. ACTA Press, 2003.

[51] E. Cuesta, C. Lubich, and C. Palencia. Convolution quadrature time discretization of fractional diffusion-wave equations. *Mathematics of Computation*, 75:673–696, January 2006.

[52] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, 1992.

[53] I. Daubechies and G. Teschke. Variational image restoration by means of wavelets: Simultaneous decomposition, deblurring and denoising. *Applied and Computational Harmonic Analysis*, 19(1):1–16, 2005.

[54] P. L. Davies and A. Kovac. Local extremes, runs, strings and multiresolution. *Annals of Statistics*, 29:1–65, 2001.

[55] R. Deriche and O. Faugeras. Les EDP en traitement des images et vision par ordinateur. *Traitment du Signal*, 13(6), 1996.

[56] S. Didas. Higher order variational methods for noise removal in signals and images. Diplomarbeit, Fachrichtung Mathematik, Universität des Saarlandes, Saarbrücken, Germany, April 2004.

[57] S. Didas, B. Burgeth, A. Imiya, and J. Weickert. Regularity and scale-space properties of fractional high order linear filtering. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*, pages 13–25. Springer, Berlin, 2005.

[58] S. Didas, P. Mrázek, and J. Weickert. Energy-based image simplification with nonlocal data and smoothness terms. In A. Iske and J. Levesley, editors, *Algorithms for Approximation – Proceedings of the 5th International Conference, Chester, July 2005*, pages 51–60. Springer, 2007.

[59] S. Didas, S. Setzer, and G. Steidl. Combined ℓ_2 data and gradient fitting in conjunction with ℓ_1 regularization. *Advances in Computational Mathematics*, 2007, in print.

[60] S. Didas and J. Weickert. From adaptive averaging to accelerated nonlinear diffusion filtering. In K. Franke, K.-R. Müller, B. Nicolay, and R. Schäfer, editors, *Pattern Recognition*, volume 4174 of *Lecture Notes in Computer Science*, pages 101–110. Springer, Berlin, September 2006.

[61] S. Didas and J. Weickert. Combining curvature motion and edge-preserving denoising. In F. Sgallari, A. Murli, and N. Paragios, editors, *Scale Space and Variational Methods in Computer Vision*, volume 4485 of *Lecture Notes in Computer Science*, pages 568–579. Springer, Berlin, 2007.

[62] S. Didas and J. Weickert. Integrodifferential equations for continuous multiscale wavelet shrinkage. *Inverse Problems and Imaging*, 1(1):47–62, February 2007.

[63] S. Didas, J. Weickert, and B. Burgeth. Stability and local feature enhancement of higher order nonlinear diffusion filtering. In W. Kropatsch, R. Sablatnig, and A. Hanbury, editors, *Pattern Recognition*, volume 3663 of *Lecture Notes in Computer Science*, pages 451–458. Springer, 2005.

[64] A. Dold and B. Eckmann, editors. *Fractional Calculus and Its Applications*, volume 457 of *Lecture Notes in Mathematics*. Springer, Berlin, 1975.

[65] D. L. Donoho and I. M. Johnstone. Ideal spatial adaption by wavelet shrinkage. *Biometrica*, 81(3):425–455, 1994.

[66] D. L. Donoho and I. M. Johnstone. Minimax estimation via wavelet shrinkage. *The Annals of Statistics*, 26(3):879–921, 1998.

[67] R. Duits, M. Felsberg, L. Florack, and B. Platel. α scape spaces on a bounded domain. In L. D. Griffin and M. Lillholm, editors, *Scale-Space 2003*, volume 2695 of *Lecture Notes in Computer Science*, pages 494–510. Springer, Berlin, 2003.

[68] R. Duits, M. Felsberg, G. Granlund, and B. ter Haar Romeny. Image analysis and reconstruction using a wavelet transform constructed from a reducible representation of the Euclidean motion group. *International Journal of Computer Vision*, 72(1):79–102, April 2007.

[69] R. Duits, L. Florack, J. de Graaf, and B. Ter Haar Romeny. On the axioms of scale space theory. *Journal of Mathematical Imaging and Vision*, 20:267–298, 2004.

[70] S. Durand and J. Froment. Reconstruction of wavelet coefficients using total-variation minimization. *SIAM Journal on Scientific Computing*, 24(5):1754–1767, 2003.

[71] S. Durand and M. Nikolova. Restoration of wavelet coefficients by minimizing a specially designed objective function. In O. Faugeras and N. Paragios, editors, *Proc. Second IEEE Workshop on Geometric and Level Set Methods in Computer Vision*, pages 145–152, Nice, France, October 2003. INRIA.

[72] M. Elad. On the origin of the bilateral filter and ways to improve it. *IEEE Transactions on Image Processing*, 11(10):1141–1151, October 2002.

[73] S. Esedoglu. An analysis of the Perona-Malik scheme. *Communications in Pure and Applied Mathematics*, 54:1442–1487, 2001.

[74] S. Esedoglu. Stability properties of the Perona-Malik scheme. *SIAM Journal on Numerical Analysis*, 44(3):1297–1313, 2006.

[75] C. Feddern, J. Weickert, B. Burgeth, and M. Welk. Curvature-driven PDE methods for matrix-valued images. *International Journal of Computer Vision*, 69(1):91–103, August 2006.

[76] M. Felsberg, R. Duits, and L. Florack. The monogenic scale-space on a bounded domain and its applications. In L. D. Griffin and M. Lillholm, editors, *Scale-Space 2003*, volume 2695 of *Lecture Notes in Computer Science*, pages 209–224, 2003.

[77] M. Felsberg and G. Sommer. Scale-adaptive filtering derived from the Laplace equation. In B. Radig and S. Florczyk, editors, *Pattern Recognition*, volume 2032 of *Lecture Notes in Computer Science*, pages 95–106. Springer, 2001.

[78] M. Felsberg and G. Sommer. The monogenic scale-space: A unifying approach to phase-based image processing in scale-space. *Journal of Mathematical Imaging and Vision*, 21(1):5–26, July 2004.

[79] X. Feng and A. Prohl. Analysis of total variation flow and its finite element approximations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(3):533–556, 2003.

[80] L. Florack. *Image Structure*, volume 10 of *Computational Imaging and Vision*. Kluwer, Dordrecht, 1997.

[81] D. Gabor. Theory of communication. *Journal of the Institute for Electrical Engineering*, 93:429–457, 1946.

[82] D. Gabor. Information theory in electron microscopy. *Laboratory Investigation*, 14(6):801–807, 1965.

[83] M. Gage. Curve shortening makes convex curves circular. *Inventiones Mathematicae*, 76:357–364, 1984.

[84] M. Gage and R. S. Hamilton. The heat equation shrinking convex plane curves. *Journal of Differential Geometry*, 23:69–96, 1986.

[85] H.-Y. Gao. Wavelet shrinkage denoising using the non-negative garrote. *Journal of Computational and Graphical Statistics*, 7(4):469–488, 1998.

[86] I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, revised english edition, 1963.

[87] G. Gerig, O. Kübler, R. Kikinis, and F. A. Jolesz. Nonlinear anisotropic filtering of MRI data. *IEEE Transactions on Medical Imaging*, 11:221–232, 1992.

[88] M. Giaquinta and S. Hildebrandt. *Calculus of Variations I – The Lagrangian Formalism*. Springer, Berlin, 1996.

[89] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, second edition, 2002.

[90] R. Gorenflo. Fractional calculus: Some numerical methods. In A. Carpinteri and F. Mainardi, editors, *Fractals and Fractional Calculus in Continuum Mechanics*, pages 277–290. Springer, Wien, New York, 1997.

[91] P. J. Green. Bayesian reconstruction from emission tomography data using a modified EM algorithm. *IEEE Transactions on Medical Imaging*, 9(1):84–93, March 1990.

[92] J. B. Greer and A. L. Bertozzi. H^1 solutions of a class of fourth order nonlinear equations for image processing. *Discrete and Continuous Dynamical Systems*, 10(1 and 2), January and March 2004.

[93] Lewis D. Griffin. Histograms of infinitesimal neighbourhoods. In M. Kerckhove, editor, *Scale-Space 2001*, volume 2106 of *Lecture Notes in Computer Science*, pages 326–334. Springer and IEEE/CS, 2001.

[94] F. Guichard and J.-M. Morel. A note on two classical shock filters and their asymptotics. In M. Kerckhove, editor, *Scale-Space and Morphology in Computer Vision*, volume 2106 of *Lecture Notes in Computer Science*, pages 75–84. Springer, 2001.

[95] S. Gupta and J. Prince. Stochastic models for div-curl optical flow methods. *IEEE Signal Processing Letters*, 3(2):32–34, 1996.

[96] A. Haar. Zur Theorie der orthogonalen Funktionen-Systeme. *Mathematische Annalen*, 69(3):331–371, September 1910.

[97] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics*. Probability and Mathematical Statistics. Wiley & Sons, New York, 1986.

[98] W. Hinterberger, M. Hintermüller, K. Kunisch, M. von Oehsen, and O. Scherzer. Tube methods for BV regularization. *Journal of Mathematical Imaging and Vision*, 19:223–238, 2003.

[99] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In J. M. Combes, A. Grossmann, and P. Tchamitchian, editors, *Wavelets, Time-Frequency Methods and Phase Space*, pages 286–297. Springer, Berlin, 1989.

[100] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[101] G. Huisken. Flow by mean curvature of convex surfaces into spheres. *Journal of Differential Geometry*, 20:237–266, 1984.

[102] T. Iijima. Basic theory on normalization of pattern (in case of typical one-dimensional pattern). *Bulletin of the Electrotechnical Laboratory*, 26:368–388, 1962. (In Japanese).

[103] T. Iijima. Theory of pattern recognition. *Electronics and Communications in Japan*, pages 123–134, 1963. (In English).

[104] JPEG2000 image coding system – JPEG2000 final committee draft version 1.0. http://www.jpeg.org/public/fcd15444-1.pdf.

[105] F. Kanters, L. Florack, B. Platel, and B. ter Haar Romeny. Image reconstruction from multiscale critical points. In L. D. Griffin and M. Lillholm, editors, *Scale-Space 2003*, volume 2695 of *Lecture Notes in Computer Science*, pages 464–478. Springer, Berlin, 2003.

[106] F. Kanters, M. Lillholm, R. Duits, B. Janssen, B. Platel, L. Florack, and B. ter Haar Romeny. On image reconstruction from multiscale top points. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale-Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*, pages 431–442. Springer, Berlin, 2005.

[107] B. Kawohl and N. Kutev. Maximum and comparison principle for one-dimensional anisotropic diffusion. *Mathematische Annalen*, 311:107–123, 1998.

[108] S. L. Keeling and G. Haase. Geometric multigrid for high-order regularizations of early vision problems. *Applied Mathematics and Computation*, 184(2):536–556, January 2007.

[109] S. L. Keeling and R. Stollberger. Nonlinear anisotropic diffusion filtering for multiscale edge enhancement. *Inverse Problems*, 18:175–190, 2002.

[110] S. Kichenassamy. The Perona-Malik paradox. *SIAM Journal on Applied Mathematics*, 57(5):1328–1342, October 1997.

[111] B. B. Kimia, A. Tannenbaum, and S. W. Zucker. Toward a computational theory of shape: an overview. In O. Faugeras, editor, *Computer Vision – ECCV '90*, volume 427 of *Lecture Notes in Computer Science*, pages 402–407. Springer, Berlin, 1990.

[112] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[113] A. Kuijper. p-Laplacian driven image processing. In *Proceedings of the International Conference on Image Processing (ICIP), September 16-19, 2007*, San Antonio, Texas, 2007, to appear.

[114] J.-S. Lee. Digital image smoothing and the sigma filter. *Computer Vision, Graphics, and Image Processing*, 24:255–269, 1983.

[115] E. H. Lieb. On the lowest eigenvalue of the Laplacian for the intersection of two domains. *Inventiones Mathematicae*, 74(3):441–448, October 1983.

[116] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer, Boston, 1994.

[117] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):117–154, 1998.

[118] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

[119] M. Lindenbaum, M. Fischer, and A. Bruckstein. On Gabor's contribution to image enhancement. *Pattern Recognition*, 27(1):1–8, 1994.

[120] A. K. Louis, P. Maaß, and A. Rieder. *Wavelets*. B. G. Teubner Stuttgart, second edition, 1998.

[121] T. Lu, P. Neittaanmäki, and X.-C. Tai. A parallel splitting up method and its application to Navier-Stokes equations. *Applied Mathematics Letters*, 4(2):25–29, 1991.

[122] M. Lysaker, A. Lundervold, and X.-C. Tai. Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. *IEEE Transactions on Image Processing*, 12(12):1579–1590, December 2003.

[123] M. Lysaker and X.-C. Tai. Iterative image restoration combining total variation minimization and a second-order functional. *International Journal of Computer Vision*, 66(1):5–18, January 2006.

[124] M. Mahmoudi and G. Sapiro. Fast image and video denoising via non-local means of similar neighborhoods. *IEEE Signal Processing Letters*, 12(12):839–842, December 2005.

[125] F. Malgouyres. Mathematical analysis of a model which combines total variation and wavelet for image restoration. *Inverse Problems*, 2(1):1–10, 2002.

[126] S. Mallat. Multiresolution approximations and wavelet orthonormal bases of $\mathscr{L}^2(\mathbb{R})$. *Transactions of the Americal Mathematical Society*, 315:69–87, September 1989.

[127] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[128] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, second edition, 1999.

[129] E. Mammen and S. van de Geer. Locally adaptive regression splines. *The Annals of Statistics*, 25(1):387–413, 1997.

[130] O. L. Mangasarian and L. L. Schumaker. Discrete splines via mathematical programming. *SIAM Journal on Control and Optimization*, 9(2):174–183, May 1971.

[131] O. L. Mangasarian and L. L. Schumaker. Best summation formulae and discrete splines. *SIAM Journal on Numerical Analysis*, 10(3):448–459, June 1973.

[132] M. Mathias. Über positive Fourier-Integrale. *Mathematische Zeitschrift*, 16:101–125, 1923.

[133] B. Mathieu, P. Melchior, A. Oustaloup, and C. Ceyral. Fractional differentiation for edge detection. *Signal Processing*, 83(11):2421–2432, November 2003.

[134] Y. Meyer. *Wavelets and Operators*, volume 37 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1992.

[135] Y. Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, volume 22 of *University Lecture Series*. AMS, Providence, 2001.

[136] F. Mintzer. Filters for distrotion-free two-band multirate filter banks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(3):626–630, June 1985.

[137] P. Mrázek and J. Weickert. Rotationally invariant wavelet shrinkage. In B. Michaelis and G. Krell, editors, *Pattern Recognition*, volume 2781 of *Lecture Notes in Computer Science*, pages 156–163. Springer, Berlin, 2003.

[138] P. Mrázek and J. Weickert. From two-dimensional nonlinear diffusion to coupled Haar wavelet shrinkage. *Journal of Visual Communication and Image Representation*, 18(2):162–175, 2007.

[139] P. Mrázek, J. Weickert, and A. Bruhn. On robust estimation and smoothing with spatial and tonal kernels. In R. Klette, R. Kozera, L. Noakes, and J. Weickert, editors, *Geometric Properties for Incomplete Data*, volume 31 of *Computational Imaging and Vision*, pages 335–352. Springer, Dordrecht, 2006.

[140] P. Mrázek, J. Weickert, and G. Steidl. Diffusion-inspired shrinkage functions and stability results for wavelet denoising. *International Journal of Computer Vision*, 64(2/3):171–186, 2005.

[141] P. Mrázek, J. Weickert, G. Steidl, and M. Welk. On iterations and scales of nonlinear filters. In O. Drbohlav, editor, *Proc. of the Computer Vision*

Winter Workshop 2003, pages 61–66. Czech Pattern Recognition Society, February 2003.

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

[143] M. Nielsen, L. Florack, and R. Deriche. Regularization, scale-space and edge detection filters. *Journal of Mathematical Imaging and Vision*, 7:291–307, 1997.

[144] W. J. Niessen, B. M. ter Haar Romeny, L. M. Florack, and M. A. Viergever. A general framework for geometry-driven evolution equations. *International Journal of Computer Vision*, 21(3):187–205, 1997.

[145] M. Nikolova. Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers. *SIAM Journal on Numerical Analysis*, 40(3):965–994, 2002.

[146] M. Nikolova. A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision*, 20:99–120, 2004.

[147] N. Nordström. Biased anisotropic diffusion – a unified regularization and diffusion approach to edge detection. *Image and Vision Computing*, 8:318–327, 1990.

[148] K. B. Oldham and J. Spanier. *The Fractional Calculus*, volume 111 of *Mathematics in Science and Engineering*. Academic Press, New York and London, 1974.

[149] S. Osher and L. I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis*, 27(4):919–940, August 1990.

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

[151] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67(2):141–158, April 2006.

[152] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Computer Vision – ECCV 2006*, volume 3954 of *Lecture Notes in Computer Science*, pages 568–580. Springer, May 2006.

[153] E. J. Pauwels, L. J. Van Gool, P. Fiddelaers, and T. Moons. An extended class of scale-invariant and recursive scale space filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):691–701, July 1995.

[154] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.

[155] L. Pizarro, S. Didas, F. Bauer, and J. Weickert. Evaluating a general class of filters for image denoising. In B. K. Ersbøll and K. S. Pedersen, editors, *Image Analysis*, volume 4522 of *Lecture Notes in Computer Science*, pages 601–610. Springer, Berlin, 2007.

[156] G. Plonka and G. Steidl. A multiscale wavelet-inspired scheme for nonlinear diffusion. *International Journal of Wavelets, Multiresolution and Information Processing*, 4(1):1–21, 2006.

[157] J. Polzehl and V. Spokoiny. Adaptive weights smoothing with applications to image restoration. *Journal of the Royal Statistical Society, Series B*, 62(2):335–354, 2000.

[158] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[159] W. Rudin. *Real and Complex Analysis*. McGraw-Hill, third edition, 1986.

[160] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, second edition, 2003.

[161] P. Saint-Marc, J.-S. Chen, and G. Medioni. Adaptive smoothing: A general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):514–529, June 1991.

[162] G. Sapiro. Vector (self) snakes: a geometric framework for color, texture and multiscale image segmentation. In *Proc. 1996 IEEE International Conference on Image Processing*, volume 1, pages 817–820, Lausanne, Switzerland, September 1996.

[163] O. Scherzer. Denoising with higher order derivatives of bounded variation and an application to parameter estimation. *Computing*, 60(1):1–27, March 1998.

[164] O. Scherzer and C. Groetsch. Inverse scale space theory for inverse problems. In M. Kerckhove, editor, *Scale-Space 2001*, volume 2106 of *Lecture Notes in Computer Science*, pages 317–325. Springer and IEEE/CS, 2001.

[165] O. Scherzer and J. Weickert. Relations between regularization and diffusion filtering. *Journal of Mathematical Imaging and Vision*, 12:43–63, 2000.

[166] C. Schnörr. Unique reconstruction of piecewise smooth images by minimizing strictly convex non-quadratic functionals. *Journal of Mathematical Imaging and Vision*, 4:189–198, 1994.

[167] H. R. Schwarz. *Numerische Mathematik*. B. G. Teubner, Stuttgart, 1997.

[168] M. Smith and T. Barnwell III. Exact reconstruction techniques for tree-structures subband coders. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(3):434–441, June 1986.

[169] S. M. Smith and J. M. Brady. SUSAN – A new approach to low lewel image processing. *International Journal of Computer Vision*, 23(1):43–78, 1997.

[170] N. Sochen, R. Kimmel, and R. Malladi. A geometrical framework for low level vision. *IEEE Transactions on Image Processing*, 7(3):310–318, 1998.

[171] J. Sporring, M. Nielsen, L. Florack, and P. Johansen, editors. *Gaussian Scale-Space Theory*, volume 8 of *Computational Imaging and Vision*. Kluwer, Dordrecht, 1997.

[172] J.-L. Starck, M. Elad, and D. L. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *IEEE Transactions on Image Processing*, 14(10):1570–1582, October 2005.

[173] G. Steidl. A note on the dual treatment of higher order regularization functionals. *Computing*, 76(1–2):135–148, January 2006.

[174] G. Steidl, S. Didas, and J. Neumann. Relations between higher order TV regularization and support vector regression. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*, pages 515–527. Springer, Berlin, 2005.

[175] G. Steidl, S. Didas, and J. Neumann. Splines in higher order TV regularization. *International Journal of Computer Vision*, 70(3):241–255, July 2006.

[176] G. Steidl, J. Weickert, T. Brox, P. Mrázek, and M. Welk. On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and SIDEs. *SIAM Journal on Numerical Analysis*, 42(2):686–713, 2004.

[177] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer, New York, third edition, 2002.

[178] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, 1986.

[179] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1997.

[180] D. Suter. Motion estimation and vector splines. In *Proceedings of CVPR 1994*, pages 939–942, June 1994.

[181] M. E. Taylor. *Pseudodifferential Operators*. Princeton University Press, Princeton, New Jersey, 1981.

[182] M. E. Taylor. *Partial Differential Equations I – Basic Theory*. Springer, New York, 1996.

[183] M. E. Taylor. *Partial Differential Equations II – Qualitative Studies of Linear Equations*. Springer, New York, 1996.

[184] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Mathematics Doklady*, 4(2):1035–1038, 1963.

[185] C. Tomasi and R. Manduchi. Bilateral filtering for gray and colour images. In *Proc. of the 1998 IEEE International Conference on Computer Vision*, pages 839–846, Bombay, India, January 1998. Narosa Publishing House.

[186] L. N. Trefethen. *Spectral Methods in Matlab*. SIAM, 2001.

[187] D. Tschumperlé and R. Deriche. Diffusion PDE's on vector-valued images: Local approach and geometric viewpoint. *IEEE Signal Processing Magazine*, 19(5):16–25, 2002.

[188] D. Tschumperlé and R. Deriche. Vector-valued image regularization with PDE's: A common framework for different applications. *IEEE Transactions on Image Processing*, 27(4):1–12, April 2005.

[189] J. Tumblin and G. Turk. LCIS: A boundary hierarchy for detail-preserving contrast reduction. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 83–90. ACM Press/Addison-Wesley Publishing Co., 1999.

[190] M. Vetterli. Multidimensional subband coding: some theory and algorithms. *Signal Processing*, 6(2):97–112, April 1984.

[191] M. Vetterli. Filter banks allowing perfect reconstruction. *Signal Processing*, 10(3):219–244, April 1986.

[192] M. Vetterli and J. Kovačevic. *Wavelets and Subband Coding*. Prentice-Hall, Upper Saddle River, 1995.

[193] O. Vogel, A. Bruhn, J. Weickert, and S. Didas. Direct shape-from-shading with adaptive higher order regularisation. In F. Sgallari, A. Murli, and N. Paragios, editors, *Scale Space and Variational Methods in Computer Vision*, volume 4485 of *Lecture Notes in Computer Science*, pages 871–882. Springer, Berlin, 2007.

[194] J. B. Weaver, Y. Xu, D. M. Healy, and L. D. Cromwell. Filtering noise from images with wavelet transforms. *Magnetic Resonance in Medicine*, 21:288–295, 1991.

[195] G. W. Wei. Generalized Perona-Malik equation for image restoration. *IEEE Signal Processing Letters*, 6(7):165–167, July 1999.

[196] J. Weickert. Anisotropic diffusion filters for image processing based quality control. In A. Fasano and M. Primicerio, editors, *Proc. Seventh European Conf. on Mathematics in Industry*, pages 355–362, Stuttgart, 1994. Teubner.

[197] J. Weickert. Theoretical foundations of anisotropic diffusion in image processing. *Computing*, Suppl. 11:221–236, 1996.

[198] J. Weickert. *Anisotropic Diffusion in Image Processing*. B. G. Teubner, Stuttgart, 1998.

[199] J. Weickert. Coherence-enhancing diffusion filtering. *International Journal of Computer Vision*, 31(2/3):111–127, 1999.

[200] J. Weickert. Applications of nonlinear diffusion in image processing and computer vision. *Acta Mathematica Universitatis Comenianae*, LXX(1):33–50, 2001.

[201] J. Weickert. Coherence-enhancing shock filters. In B. Michaelis and G. Krell, editors, *Pattern Recognition*, volume 2781 of *Lecture Notes in Computer Science*, pages 1–8. Springer, Berlin, 2003.

[202] J. Weickert and B. Benhamouda. A semidiscrete nonlinear scale-space theory and its relation to the perona-malik paradox. In F. Solina, W. G. Kropatsch, R. Klette, and R. Bajcsy, editors, *Advances in Computer Vision*, pages 1–10. Springer, Wien, 1997.

[203] J. Weickert, C. Feddern, M. Welk, B. Burgeth, and T. Brox. PDEs for tensor image processing. In J. Weickert and H. Hagen, editors, *Visualization and Processing of Tensor Fields*, Mathematics and Visualization, pages 399–414. Springer, Berlin, 2006.

[204] J. Weickert, S. Ishikawa, and A. Imiya. Linear scale-space has first been proposed in Japan. *Journal of Mathematical Imaging and Vision*, 10:237–252, 1999.

[205] J. Weickert, G. Steidl, P. Mrázek, M. Welk, and T. Brox. Diffusion filters and wavelets: What can they learn from each other? In N. Paragios, Y. Chen, and O. Faugeras, editors, *Handbook of Mathematical Models in Computer Vision*, pages 3–16. Springer, New York, 2006.

[206] J. Weickert, B. M. ter Haar Romeny, and M. A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, March 1998.

[207] J. Weickert and M. Welk. Tensor field interpolation with PDEs. In J. Weickert and H. Hagen, editors, *Visualization and Processing of Tensor Fields*, Mathematics and Visualization, pages 315–325. Springer, Berlin, 2006.

[208] M. Welk, G. Steidl, and J. Weickert. Locally analytic schemes: A link between diffusion filtering and wavelet shrinkage. *Applied and Computational Harmonic Analysis*, 2007, in press.

[209] D. Werner. *Funktionalanalysis*. Springer, Berlin, third edition, 2000.

[210] G. Winkler, V. Aurich, K. Hahn, and A. Martin. Noise reduction in images: Some recent edge-preserving methods. *Pattern Recognition and Image Analysis*, 9(4):749–766, 1999.

[211] A. P. Witkin. Scale-space filtering. In *Proc. Eighth International Joint Conference on Artificial Intelligence*, volume 2, pages 945–951, Karlsruhe, Germany, August 1983.

[212] L. P. Yaroslavsky. *Digital Picture Processing*. Springer, New York, 1985.

[213] Y.-L. You and M. Kaveh. Fourth-order partial differential equations for noise removal. *IEEE Transactions on Image Processing*, 9(10):1723–1730, October 2000.

[214] E. Zeidler. *Nonlinear Functional Analysis and Applications I: Fixed-Point Theorems*. Springer, New York, 1986.