Saarland University
Faculty of Mathematics and Computer Science
Department of Computer Science

Master's Thesis

Analyzing and Extending Shock Filters

Submitted by:

Kristina Schaefer

on November 27, 2020

Supervisor
Prof. Dr. Joachim Weickert

Reviewers

Prof. Dr. Joachim Weickert

Dr. Pascal Peter

Erklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Statement

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis

Einverständniserklärung

Ich bin damit einverstanden, dass meine (bestandene) Arbeit in beiden Versionen in die Bibliothek der Informatik aufgenommen und damit veröffentlicht wird.

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken,______________________ ______________________________

(Datum/Date) (Unterschrift/Signature)

Abstract

A shock filter is a morphological filter that applies dilation and erosion, depending on a guidance term. This class of filters forms an interesting phenomenon that has not been understood very well. Moreover, the filters are very noise sensitive. Therefore, shock filters are not used often.
To improve the understanding of shock filters, we first simplify them by using a fixed guidance term. We prove that in that case all shock filters must behave equally in strictly convex and concave regions. These findings help to explain the behavior of shock filters with a non-fixed guidance term. We also use this property to explain why shock filters are well suited to deblur cartoon-like images.

The noise sensitivity of shock filters has been addressed by combining smoothing and shock filtering in two different ways: the usage of smoothing in the guidance term and the linear combination of smoothing and shock filtering. We show that presmoothing inside of the guidance term does not actually lead to a regularized shock filter, but to the creation of different structures. As a linear combination of shock filtering and smoothing, we propose the shock-diffusion filter, which is a weighted sum of shock filtering and homogeneous diffusion. Using a classical diffusivity as the weighting function leads to a behavior that is similar to nonlinear isotropic diffusion or even anisotropic diffusion, depending on the shock term. We also propose a way to modify diffusivities, such that the shock-diffusion can converge to a non-flat steady state.

The similarity of shock-diffusion and anisotropic diffusion motivates us to apply shock-diffusion to inpainting problems. We found that the best performance is achieved by using the shock term that Weickert uses in his Coherence Enhancing Shock Filter [32]. The comparison of the inpainting results of the shock-diffusion operator and results produced by EED implies that shock-diffusion can achieve similar quality to EED and it can create even sharper edges than EED.

Acknowledgments

First, I want to thank Prof. Joachim Weickert for introducing me to the field of image processing and supervising both my bachelor's and master's thesis. He has always been open for any questions and discussions and provided test images and source code for several experiments.

Next, I want to thank Dr. Pascal Peter for being my second reviewer.

Moreover, I thank Steven Dlucik for proofreading my thesis.

Finally, I want to thank my family and friends for their unconditional support during my studies.

Contents

Chapter 1

Introduction

Image processing based on partial differential equations (PDEs) has been successful in numerous different fields including denoising, image compression and image segmentation. One morphological class of PDE-based filters is given by shock filters. Like many morphological methods, shock filters are based on the operations dilation and erosion. Based on a second derivative operator, a guidance term determines whether dilation or erosion is applied. This combination of dilation and erosion leads to sharpening the edges defined by the guidance term and eventually to a segmented and, potentially, non-flat steady state.

With the goal of deblurring images, Kramer et al. gave the first discrete definition of a shock filter in 1975 [16]. 15 years later, the first PDE-formulation was given by Osher and Rudin in 1990 [21]. To further advance shock filtering, it was combined with smoothing operators in different ways. Weickert [32] and Alavarez and Mazorra [2] embedded the smoothing operation into the definition of the shock filter. Another popular approach is coupling the PDEs of the shock-filter and the smoothing operator [2, 15, 12, 11, 3, 38].

In this thesis, we contribute to the class of shock filters in three ways. First, we investigate the role of the guidance term and show that shock filters with different guidance terms often behave similar. We show that, for a fixed guidance term, all shock filter behave equally in strictly convex and strictly concave regions, independently of the second derivative operator that guides the evolution.

Secondly, we investigate the combination of homogeneous diffusion and shock filtering. Diffusion is often embedded into the shock filter definition as Gaussian-smoothing of the second derivative operator. Using different experiments, we evaluate the behavior of such a shock filter with presmoothing. By using a weighted sum of shock and diffusion term, we create a shock-

diffusion filter than behaves similar to methods from diffusion, like nonlinear isotropic diffusion or Edge Enhancing Diffusion (EED). We show that the shock-diffusion filter can perform denoising and segmentation in a quality similar to well-proven methods like EED. Moreover, we propose a weighting function that even yields non-flat steady states for the aforementioned shock-diffusion filter.
Finally, we investigate some applications of the different shock filter variants in this thesis. We show that pure shock filters perform well for deblurring, especially, cartoon-like images. Since our shock-diffusion filter yields results that are comparable to some diffusion methods, we also evaluate its performance in inpainting.

1.1 Structure of this Thesis

In the remainder of this Chapter, we briefly recap basic information that is used throughout this thesis. In Chapter 2, we explain the basic ideas behind shock filters and introduce the shock filtering framework that we are working with. Our shock filter definition is purposely relatively open, which allows us to look into different derivative operators and guidance terms that determine the behavior of the shock filter. In Chapter 3, we introduce the derivative operators that are relevant for the remaining chapters. In Chapter 4 and 5, we look into different options for the guidance term of shock filters and explain the resulting behavior. As shock filters were originally created for the purpose of deblurring images, we look into deblurring with shock filters in Chapter 6. The shock filter definitions of Alvarez and Mazorra [2] and Weickert [32] use a Gaussian-smoothing in the computation of the guidance term. We investigate the effect of this smoothing operation in Chapter 7. In Chapter 8, we propose a shock-diffusion filter that combines diffusion and shock filtering. We explain its behavior and compare its performance to some pure diffusion methods. In Chapter 9.1, we evaluate the performance of the mentioned shock-diffusion filter in several inpainting tasks. We compare the produced results to the performance of Edge Enhancing Diffusion.

1.2 Fundamental Information

In this thesis, a number of different methods from image processing are used. This section provides the basic definitions and the notation used throughout this thesis.

1.2.1 Signals and Images

In general, we treat signals and images as mappings from a coordinate to a value. A n-dimensional *continuous signal* is a function that maps a real-valued, n-dimensional vector to a real number:

$$f : \Gamma \subset \mathbb{R}^n \to \mathbb{R} \tag{1.1}$$

As a special case of a continuous signal, a *continuous image* maps a coordinate from the rectangular *image domain* Γ to a real number:

$$f : \Gamma \subset \mathbb{R}^n \to \mathbb{R}, \qquad \text{where } \Gamma = [a, b] \times [c, d] \tag{1.2}$$

Signals, as well as images often originate from the real world and are then digitized using measurement devices, such as light sensors. During this process, such an originally continuous function is measured at several positions. This procedure is often referred to as sampling. For images, the positions are typically defined by a rectangular grid with the resolution $n_x \times n_y$. The distance between measuring positions is determined by the grid sizes h_x and h_y.
Sampling a continuous signal results in its discretization. An n-dimensional *discrete signal* is defined by

$$f : \Omega \subset \mathbb{N}^n \to \mathbb{N} \tag{1.3}$$

Similarly, we define *discrete* or *digital image* as a mapping from a coordinate to a grey value as

$$f : \Omega \subset \mathbb{N}^2 \to \mathbb{N} \qquad \text{where } \Omega = \{1 \ldots n_x\} \times \{1 \ldots n_y\} \tag{1.4}$$

We will refer to such a coordinate as pixel. The grey value of an image f in a pixel (i, j) is denoted $u_{i,j}$.
The values of the images in this thesis are typically grey values within the range of $[0 \ldots 255]$.

Since our filters use several iterations, we also need a discretization in the time domain. The sampling distance τ in the time domain is called time step size. The image u in a pixel (i, j) at the time k is denoted $u^k_{i,j}$.

Rescaling

In some examples, we use two dimensional functions that originally have values outside the range of $[0 \ldots 255]$. In order to make these values visible, we apply affine rescaling. For the original image f the rescaled version u is given by

$$u_{i,j} = 255 \frac{f_{i,j} - f_{min}}{f_{max} - f_{min}} \tag{1.5}$$

where f_{min} is the minimal value of f and f_{max} the maximal value.

1.2.2 Derivatives

The topics covered in this thesis are heavily dependent on various derivative operators and modeling of image filter as *partial differential equations* (PDE). So naturally, derivatives will appear often.

In general, derivatives of a function measure the change of the function's values. *Directional derivatives* measure the change in a certain direction. We denote the directional derivative of $f : \mathbb{R}^n \to \mathbb{R}$ in a direction $\boldsymbol{v} \in \mathbb{R}^n, |\boldsymbol{v}| = 1$ at a point $\boldsymbol{x}$ as:

$$f_{\boldsymbol{v}} = \partial_{\boldsymbol{v}} f = \lim_{h \to 0} \frac{f(\boldsymbol{x} + h\boldsymbol{v}) - f(\boldsymbol{x})}{h} \tag{1.6}$$

The i-th *partial derivative* of a function computes the derivative of f with respect to the i-th argument of the function. One can view it as a directional derivative in direction of the i-th coordinate axis of the function. We denote the i-th partial derivative as

$$f_{\boldsymbol{x_i}} = \partial_{\boldsymbol{x_i}} f = \lim_{h \to 0} \frac{f(x_1, x_2, \ldots, x_i + h, x_{i+1}, \ldots, x_n) - f(\boldsymbol{x})}{h} \tag{1.7}$$

If this limit exists and is continuous for all arguments of f, then the function is one time differentiable. For the sake of readability, we assume that all used functions are sufficiently often differentiable in the remainder of this thesis. The *gradient* $\boldsymbol{\nabla}$ of a function f is the vector that points in the direction of the steepest ascent of f. It is denoted by

$$\boldsymbol{\nabla} f = \begin{pmatrix} f_{\boldsymbol{x_1}} \\ f_{\boldsymbol{x_2}} \\ \vdots \\ f_{\boldsymbol{x_n}} \end{pmatrix} \tag{1.8}$$

The *gradient magnitude* corresponds to the rate of the increase in direction of the gradient. It is defined by

$$|\boldsymbol{\nabla} f| = \sqrt{f_{\boldsymbol{x_1}}^2 + f_{\boldsymbol{x_2}}^2 + \cdots + f_{\boldsymbol{x_n}}^2} \tag{1.9}$$

The gradient can be used to express the first directional derivative in a normalized direction $\boldsymbol{v}$:

$$f_{\boldsymbol{v}} = \boldsymbol{v}^T \boldsymbol{\nabla} f(\boldsymbol{x}) \tag{1.10}$$

The *Hessian* $\mathcal{H}(f)$ is the matrix with the second order partial derivatives of a function f as entries. For a two dimensional function it looks as follows:

$$\begin{aligned} \mathcal{H}(f) &= (h_{i,j}) \\ h_{i,j} &= (f(\boldsymbol{x}))_{x_i,x_j} \end{aligned} \tag{1.11}$$

The Hessian can be used to express second directional derivatives in a normalized direction $\boldsymbol{v}$ as follows:

$$\partial_{vv} f = \boldsymbol{v}^T \mathcal{H} \boldsymbol{v} \tag{1.12}$$

The definiteness of the Hessian is used to determine if a point $\boldsymbol{x}$, for which $\boldsymbol{\nabla} u = \mathbf{0}$ holds, is a maximum, minimum or neither. The Hessian is

- positive definite, if $\boldsymbol{v}^T \mathcal{H} \boldsymbol{v} > 0 \ \forall \boldsymbol{v} \in \mathbb{R}^n, \boldsymbol{v} \neq \mathbf{0}$,
- positive semi- definite, if $\boldsymbol{v}^T \mathcal{H} \boldsymbol{v} \geq 0 \ \forall \boldsymbol{v} \in \mathbb{R}^n$,
- negative definite, if $\boldsymbol{v}^T \mathcal{H} \boldsymbol{v} < 0 \ \forall \boldsymbol{v} \in \mathbb{R}^n, \boldsymbol{v} \neq \mathbf{0}$,
- negative semi-definite, if $\boldsymbol{v}^T \mathcal{H} \boldsymbol{v} \leq 0 \ \forall \boldsymbol{v} \in \mathbb{R}^n$.

A point $\boldsymbol{x}$ is a local minimum, if $\boldsymbol{\nabla} u = \mathbf{0}$ and if the $\mathcal{H}(f)$ is positive definite in $\boldsymbol{x}$. It is a local maximum, if the Hessian is negative definite.
The Hessian can also be used for the characterization of the convexity and concavity of functions. Let $R \subset \mathbb{R}^n$ be a a open and convex set and the function $f \in C^2(R)$ be at least two times differentiable. f is convex on R, if the Hessian is positive semi-definite on R and strictly convex if the Hessian is positive definite on R. Analogously, the function f is concave on R, if the Hessian is negative semi-definite on R and strictly concave if it is negative definite on R.

Approximating Derivatives

Since we actually work with discrete images, we need to use approximations for the derivatives. Such approximations are given by *Finite Differences*.

For $u : D \subset \mathbb{R} \to \mathbb{R}$ and sampling points $x_i \in D$, that are sampled with the grid size $h \geq 0$, typical operators that approximate the first derivative are *forward*, *backward* and *central difference*:

$$\text{forward difference: } (u_x)_i \approx \frac{u_{i+1} - u_i}{h} \tag{1.13}$$

$$\text{backward difference: } (u_x)_i \approx \frac{u_i - u_{i-1}}{h} \tag{1.14}$$

$$\text{central difference: } (u_x)_i \approx \frac{u_{i+1} - u_{i-1}}{2h} \tag{1.15}$$

The second derivative is often approximated by a central difference:

$$(u_{xx})_i \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \tag{1.16}$$

The approximation order of the forward and backward difference is $\mathcal{O}(h)$. The central differences give a higher accuracy with the approximation order of $\mathcal{O}(h^2)$.

1.2.3 Structure Tensor

1987, Förstner and Gülch introduced the structure tensor [10]. It can be used to analyze the local structure of an image. In general, it averages information on the first derivatives in x and y direction within a certain region, determined by Gaussian convolution. It can be defined as follows:

$$\boldsymbol{J}_\rho(\boldsymbol{\nabla} u) = K_\rho * (\boldsymbol{\nabla} u \boldsymbol{\nabla} u^T) = K_\rho * \begin{pmatrix} u_x^2 & u_x u_y \\ u_x u_y & u_y^2 \end{pmatrix} = \begin{pmatrix} j_{1,1} & j_{1,2} \\ j_{1,2} & j_{2,2} \end{pmatrix} \tag{1.17}$$

where K_ρ is a Gaussian. Its standard deviation that is called *integration scale*. The integration scale determines the locality of the structure tensor [33].

The eigenvectors of the structure tensor are the directions of the smallest and largest local contrast. The corresponding eigenvalues specify the size of the local contrast in these two directions. They are defined as

$$\begin{aligned} \lambda_1 &= j_{1,1} - j_{2,2} + \sqrt{(j_{1,1} - j_{2,2})^2 + 4j_{1,2}^2} \\ \lambda_2 &= j_{1,1} - j_{2,2} - \sqrt{(j_{1,1} - j_{2,2})^2 + 4j_{1,2}^2} \end{aligned} \tag{1.18}$$

1.2.4 Diffusion

In several parts of this thesis, we use diffusion filters or compare the proposed filters to diffusion. Diffusion in image processing is inspired by the physical diffusion process. In physics, diffusion is a movement that equalizes concentration differences between substances. Once the concentration difference is equalized, the movement stops and the steady state is reached.
Mathematically, the process can be described by the following PDE:

$$u_t = \operatorname{div}(\boldsymbol{D} \boldsymbol{\nabla} u) \tag{1.19}$$

where u is the function that describes the concentrations. The diffusion tensor $\boldsymbol{D}$ steers the diffusion.
There are two important properties of diffusion. A diffusion process is linear, if it acts equally strong at all locations. It is isotropic, if it is equally strong in all directions.

In image processing, the concentration difference corresponds to the difference between the grey values of an image. We mainly use three diffusion filters, that are described briefly in the following paragraphs. For a more detailed explanaition on diffusion filters, we refer to [30].

Homogeneous Diffusion

Homogeneous diffusion acts linear and isotropic. Therefore it is also referred to as isotropic linear diffusion. The following initial value problem describes the filter mathematically for the initial image f:

$$\begin{aligned} u_t &= \operatorname{div}(\boldsymbol{I}\boldsymbol{\nabla} u) = \Delta u, && \text{on } \Gamma \\ u(0) &= f, && \text{on } \Gamma \\ \partial_n u &= 0, && \text{on } \partial\Gamma \times (0,\infty) \end{aligned} \tag{1.20}$$

$\boldsymbol{I}$ is the identity matrix. At the image boundary, reflecting boundary conditions are used.
It can be shown [35], that this PDE has the unique solution

$$u(\boldsymbol{x},t) = \begin{cases} f(\boldsymbol{x}), & t = 0 \\ (K_{\sqrt{2t}} * f)(\boldsymbol{x}), & t > 0 \end{cases} \tag{1.21}$$

for the Gaussian $K_{\sqrt{2t}}$ of standard deviation $\sqrt{2t}$. Thus, the result of homogeneous diffusion filtering at a stopping time t is related to convolution with a Gaussian of the standard deviation $\sigma = \sqrt{2t}$.

Nonlinear Isotropic Diffusion

Perona and Malik proposed the concept of nonlinear isotropic diffusion in [22]. The strength of the diffusion is scaled by a function called diffusivity. Diffusivities are positive, infinitely often differentiable functions. Typically, they are used to scale down the strength of the diffusion at edges of objects. To achieve that effect, the diffusivity is chosen as a function that is decreasing in the squared gradient magnitude.
Nonlinear isotropic diffusion of the initial image f is described by the following initial value problem:

$$u_t = \operatorname{div}(g(|\boldsymbol{\nabla} u|^2)\boldsymbol{\nabla} u), \qquad \text{on } \Gamma$$

$$\begin{aligned} u(0) &= f, && \text{on } \Gamma \\ \partial_n u &= 0, && \text{on } \partial\Gamma \times (0, \infty) \end{aligned} \tag{1.22}$$

where g is the diffusivity.

Catté et al. [7] introduced the concept of regularized nonlinear isotropic diffusion by applying Gaussian-smoothing to the image before computing the squared gradient magnitude:

$$u_t = \operatorname{div}(g(|\boldsymbol{\nabla} u_\sigma|^2)\boldsymbol{\nabla} u) \tag{1.23}$$

where $u_\sigma = u * K_\sigma$ for the Gaussian K_σ with standard deviation σ.

Edge Enhancing Diffusion

Edge Enhancing Diffusion (EED) was proposed by Weickert in [29]. This type of diffusion acts anisotropic and nonlinear. Its mathematical description is given by

$$\begin{aligned} u_t &= \operatorname{div}(\boldsymbol{D}\boldsymbol{\nabla} u), && \text{on } \Gamma \\ u(0) &= f, && \text{on } \Gamma \\ \partial_n u &= 0, && \text{on } \partial\Gamma \times (0, \infty) \end{aligned} \tag{1.24}$$

where $\boldsymbol{D}$ is the diffusion tensor. The diffusion tensor is constructed using eigenvectors $\boldsymbol{v_1} || \boldsymbol{\nabla} u_\sigma$ and $\boldsymbol{v_2} \perp \boldsymbol{\nabla} u_\sigma$ and the eigenvalues $\lambda_1 = g(|\boldsymbol{\nabla} u_\sigma|^2)$ and $\lambda_2 = 1$. This choice leads to the process smoothing along edges and in regions, while only little diffusion happens across edges.

1.2.5 Morphology

Morphology is a successful class of image processing methods that mainly uses shape analysis. It was founded by Serra and Matheron around 1965 [28].

Most morphological operators use the operations *dilation* and *erosion*. For a function $f : \mathbb{R}^n \to \mathbb{R}$ and a *structuring element* $S \subset \mathbb{R}^n$, the dilation of f with S results in the supremum, and the erosion in the infimum, of f within S. Dilation is denoted

$$(f \oplus S)(\boldsymbol{x}) = \sup\{f(\boldsymbol{x} - \boldsymbol{y}) | \boldsymbol{y} \in S\} \tag{1.25}$$

and erosion

$$(f \ominus S)(\boldsymbol{x}) = \inf\{f(\boldsymbol{x} + \boldsymbol{y}) | \boldsymbol{y} \in S\} \tag{1.26}$$

Dilation and erosion can be formulated as PDEs. Dilation and erosion of an initial image f can be described using the following initial value problem [24, 35]

$$
\begin{aligned}
u_t &= \pm\beta|\boldsymbol{\nabla} u|, && \text{on } \Gamma \\
u(0) &= f, && \text{on } \Gamma \\
\partial_n u &= 0, && \text{on } \partial\Gamma \times (0,\infty)
\end{aligned}
\tag{1.27}
$$

where the speed function β describes the structuring element. If β is constant then the structuring element is a disk [24]. For dilation the $+$ sign s used and for erosion the $-$ sign.

To apply dilation and erosion to a digital image, we use the Rouy–Tourin upwind scheme [23] that discretizes the PDEs of dilation and erosion. Dilation with a disk-shaped structuring element is discretized as follows:

$$
\begin{aligned}
\frac{u_{i,j}^{k+1} - u_{i,j}}{\tau} = \Bigg(&\max\left\{0, \frac{u_{i+1,j}^k - u_{i,j}^k}{h_x}, \frac{u_{i-1,j}^k - u_{i,j}^k}{h_x}\right\}^2 \\
&+ \max\left\{0, \frac{u_{i,j+1}^k - u_{i,j}^k}{h_y}, \frac{u_{i,j-1}^k - u_{i,j}^k}{h_y}\right\}^2\Bigg)^{\frac{1}{2}}
\end{aligned}
\tag{1.28}
$$

Erosion is discretized as follows:

$$
\begin{aligned}
\frac{u_{i,j}^{k+1} - u_{i,j}}{\tau} = -\Bigg(&\max\left\{0, \frac{u_{i,j}^k - u_{i+1,j}^k}{h_x}, \frac{u_{i,j}^k - u_{i-1,j}^k}{h_x}\right\}^2 \\
&+ \max\left\{0, \frac{u_{i,j}^k - u_{i,j+1}^k}{h_y}, \frac{u_{i,j}^k - u_{i,j-1}^k}{h_y}\right\}^2\Bigg)^{\frac{1}{2}}
\end{aligned}
\tag{1.29}
$$

Chapter 2

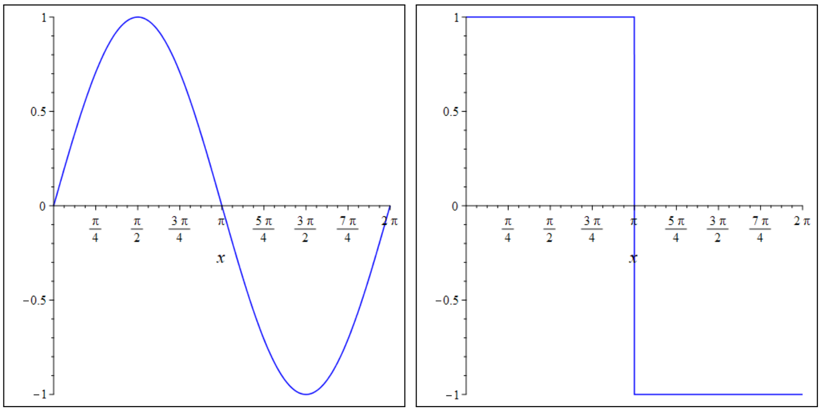
Shock Filtering Framework

In 1975, the first discrete formulation of a shock filter was given by Kramer and Bruckner in [16]. The goal of the original filter was sharpening blurred edges. To achieve that effect, pixels on the bright side of the edge need to become brighter and pixels on the darker side darker. Applying dilation on the brighter side and erosion on the darker side leads to such a sharpening effect. So the basic idea behind shock filtering is to locally apply dilation or erosion depending on the image structure.
In this chapter, we explain how the adaptation to the image structure can be realized and introduce the general PDE formulation of shock filtering that is used in this thesis. First, we give an intuition to shock filtering by explaining the one dimensional case. Afterwards, we explain how these ideas are translated to higher dimensional signals. Based on that, we define our shock filter model.

2.1 1D Case

Whether the value in a point should increase or decrease can be determined by inspecting the environment of the point. If the point lies within the influence zone of a local maximum, then it should be increased. Values in the influence zone of a minimum should be decreased. The idea becomes clear by looking at the example in Figure 2.1, which depicts the sin function. To sharpen the "edge" in the point $x = \pi$, values left of π need to be increased and values on the right need to be decreased. After the application of the shock filter, the signal is divided into two sections with the values -1 and 1.

For a one dimensional continuous signal u, a point x_0 is located within the influence zone of a maximum if $u_{xx}(x_0) < 0$ and it is located within the influence zone of a minimum if $u_{xx}(x_0) > 0$. In the first case, x_0 is located



(a) $u(x) = \sin(x)$ (b) Result of shock filter

Figure 2.1: Shock filtering a one dimensional sinus function.

within a strictly concave region and in the latter in a strictly convex region. So in a strictly concave region, the shock filter should perform dilation and in a strictly convex region it should apply erosion. Using the PDE formulations of dilation and erosion, this can be embedded in a PDE as follows:

$$\begin{aligned}
u_t &= -\operatorname{sgn}(u_{xx})|u_x| = \begin{cases} |u_x| & u_{xx} < 0 \\ -|u_x| & u_{xx} > 0\,, \\ 0 & \text{else} \end{cases} && \text{on } \Gamma \\
u(0) &= f, && \text{on } \Gamma \\
u_x &= 0, && \text{on } \partial\Gamma \times (0, \infty) \qquad (2.1)
\end{aligned}$$

2.1.1 Discrete Models and Properties

Semidiscrete Case

Equation 2.1 can be discretized with the Rouy-Tourin upwind scheme. Welk et al. use this space-discretization in [37] in their semidiscrete model. The second derivative is discretized using the central difference for second order derivative:

$$\dot{u}_i = \begin{cases} \max\left(\frac{u_{i+1}-u_i}{h}, \frac{u_{i-1}-u_i}{h}, 0\right), & 0 > \frac{u_{i+1}-2u_i+u_{i-1}}{h^2} \\ -\max\left(\frac{u_i-u_{i+1}}{h}, \frac{u_i-u_{i-1}}{h}, 0\right), & 0 < \frac{u_{i+1}-2u_i+u_{i-1}}{h^2} \\ 0, & else \end{cases} \qquad (2.2)$$

$$u_i(0) = f_i \qquad (2.3)$$

They show that this model satisfies the following properties:

1. **Existence of a unique solution**: There is a unique solution for the problem described in Eq. 2.2. The solution preserves convexity and concavity, as well as maxima and minima.

2. **Maximum-minimum principle**: For all $t \geq 0$ and points i, $\min_j(f_j) \leq u_i(t) \leq \max_j(f_j)$ holds.

3. l_∞**-stability**: The solution of the system depends l_∞-continuously on a neighborhood of the initial signal f.

4. **Total variation**: The total variation is preserved, if the total variation of the initital signal f is finite.

5. **Steady State**: The steady state is given by piecewise constant signal with discontinuities (or shocks) at maxima of the first derivative of the original signal. If the first derivative has maxima, the steady state is not flat.

Fully discrete Case

Using forward differences in time, the scheme from Eq. 2.2 can be converted to a fully discrete scheme:

$$\frac{u_i^{k+1} - u_i^k}{\tau} = \begin{cases} \max\left\{\frac{u_{i+1}^k - u_i^k}{h}, \frac{u_{i-1}^k - u_i^k}{h}, 0\right\}, & 0 > \frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{h^2} \\ -\max\left\{\frac{u_i^k - u_{i+1}^k}{h}, \frac{u_i^k - u_{i-1}^k}{h}, 0\right\}, & 0 < \frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{h^2} \\ 0, & \text{else} \end{cases} \tag{2.4}$$

$$u_i^0 = f_i \tag{2.5}$$

where f is the initial signal. Welk et al. show in [37] that for $\tau < \frac{1}{2}$ the fully discrete scheme satisfies the same properties as the semidiscrete scheme.

2.2 Higher Dimensional Case

For higher dimensional signals such as images, the construction and properties of shock filters are more complex as in the one dimensional case. The general idea remains: Te filter should perform dilation around maxima and erosion around minima and thus segment the initial image into different regions. In the higher dimensional case, there are several derivative operators that can be used for determining if a pixel is in the influence zone of a maximum/minimum. We cover some of the possible operators in Chapter 3.

For an initial image f, the PDE that describes shock filtering for images can be defined as follows:

$$\begin{aligned} u_t &= -s(Lv)|\boldsymbol{\nabla} u|, && \text{on } \Gamma \\ u(0) &= f, && \text{on } \Gamma \\ \partial_n u &= 0, && \text{on } \partial\Gamma \times (0, \infty) \end{aligned} \tag{2.6}$$

Lv is a second derivative of the image v, which is not necessarily equal to the evolving image u. The function $s : \mathbb{R} \to \mathbb{R}$ acts as the speed function for dilation and erosion, while also behaving like the sign function. For s, $\operatorname{sgn}(x) = \operatorname{sgn}(s(x))$ must hold. Furthermore, we require $-1 \leq s(x) \leq 1$. We mostly use $s = \operatorname{sgn}$ throughout the thesis. Since the sign of the second derivative Lv guides the operations of the shock filter, we refer to Lv as the guidance term and to v as the guidance image.

2.2.1 Discretization

To apply the proposed shock filtering framework to discrete images, a discrete formulation of the filter class is needed. For this purpose, we first present a space-discrete scheme. Afterwards, we introduce the the fully discrete case.

Space-discrete Case

Aside from the derivative term, the shock filter essentially consists of the morphological operations dilation and erosion. This can also be seen by rewriting Equation (2.6):

$$u_t = -s(Lv)|\boldsymbol{\nabla} u| = \begin{cases} |\boldsymbol{\nabla} u||s(Lu)|, & Lv < 0 \\ -|\boldsymbol{\nabla} u||s(Lu)|, & Lv > 0 \\ 0, & else \end{cases} \tag{2.7}$$

Like for the one dimensional case, the space discretization can be constructed using the Roy-Tourin upwind scheme [23] for dilation and erosion. Using this scheme leads to the following discrete cases. For $(Lv)_{i,j} < 0$:

$$\begin{aligned} u_t = -s_{i,j}\Bigg(&\max\left\{0, \frac{u_{i+1,j} - u_{i,j}}{h_x}, \frac{u_{i-1,j} - u_{i,j}}{h_x}\right\}^2 \\ &+ \max\left\{0, \frac{u_{i,j+1} - u_{i,j}}{h_y}, \frac{u_{i,j-1} - u_{i,j}}{h_y}\right\}^2\Bigg)^{\frac{1}{2}} \end{aligned} \tag{2.8}$$

and for $(Lv)_{i,j} > 0$:

$$u_t = -s_{i,j}\Bigg(\max\Big\{0, \frac{u_{i,j} - u_{i+1,j}}{h_x}, \frac{u_{i,j} - u_{i-1,j}}{h_x}\Big\}^2 + \max\Big\{0, \frac{u_{i,j} - u_{i,j+1}}{h_y}, \frac{u_{i,j} - u_{i,j-1}}{h_y}\Big\}^2\Bigg)^{\frac{1}{2}} \tag{2.9}$$

For $(Lv)_{i,j} = 0$:

$$u_t = 0 \tag{2.10}$$

Fully discrete case

The time discretization is performed using forward difference. Similar to the sampling distances h_x and h_y, we use the time step size τ to express the distance between to points in time k and $k+1$. The fully discrete scheme is then given by the following cases. For $(Lv)^k_{i,j} < 0$:

$$\frac{u^{k+1}_{i,j} - u_{i,j}}{\tau} = -s_{i,j}\Bigg(\max\Big\{0, \frac{u^k_{i+1,j} - u^k_{i,j}}{h_x}, \frac{u^k_{i-1,j} - u^k_{i,j}}{h_x}\Big\}^2 + \max\Big\{0, \frac{u^k_{i,j+1} - u^k_{i,j}}{h_y}, \frac{u^k_{i,j-1} - u^k_{i,j}}{h_y}\Big\}^2\Bigg)^{\frac{1}{2}} \tag{2.11}$$

for $(Lv)^k_{i,j} > 0$:

$$\frac{u^{k+1}_{i,j} - u_{i,j}}{\tau} = -s_{i,j}\Bigg(\max\Big\{0, \frac{u^k_{i,j} - u^k_{i+1,j}}{h_x}, \frac{u^k_{i,j} - u^k_{i-1,j}}{h_x}\Big\}^2 + \max\Big\{0, \frac{u^k_{i,j} - u^k_{i,j+1}}{h_y}, \frac{u^k_{i,j} - u^k_{i,j-1}}{h_y}\Big\}^2\Bigg)^{\frac{1}{2}} \tag{2.12}$$

and for $(Lv)^k_{i,j} = 0$

$$\frac{u^{k+1}_{i,j} - u_{i,j}}{\tau} = 0 \tag{2.13}$$

The Rouy-Tourin scheme fulfills a discrete minimum-maximum principle if the time step size τ satisfies $\sqrt{\left(\frac{\tau}{h_x}\right)^2 + \left(\frac{\tau}{h_y}\right)^2} \leq 1$ Since $|s| \leq 1$ holds, the minimum-maximum principle from the Rouy-Tourin scheme applies to the given fully discrete shock filter definition under the same conditions.

2.3 Related Work

Ever since their first appearence in 1975, several different shock filter definitions have been proposed. The first discrete definition given by Kramer and Bruckner [16] can be written as follows:

$$u^{k+1} = \begin{cases} u^k \oplus N(\boldsymbol{x}), & \mathcal{NLL}(x) \leq 0 \\ u^k \ominus N(\boldsymbol{x}), & else \end{cases} \tag{2.14}$$

where $N(\boldsymbol{x})$ denotes the structuring element. In their experiments, they used a disk shaped structuring element of radius 1. For the derivative operator, they use the Morphological Laplacian $\mathcal{NLL}$. A discrete formulation was first proposed by van Vliet et al. [17] in 1989 and later a continous definition was given by Maragos in 2005 [19]. The Morphological Laplacian is a morphological second derivative operator that approximates the second derivative in gradient direction if a disk shaped structuring element is used [25].

In 1990, Osher and Rudin gave the first PDE formulation of shock filters [21]. In their definition, they use the second derivative in direction of the image gradient as the second derivative operator and $v = u$ as the guidance image. Using the sgn function as the speed function yields the following definition

$$u_t = -\mathrm{sgn}(u_{\eta\eta})|\boldsymbol{\nabla} u| \tag{2.15}$$

where $\boldsymbol{\eta} = \frac{\boldsymbol{\nabla} u}{|\boldsymbol{\nabla} u|}$.

Alvarez and Mazorra modified the scheme of Osher and Rudin by Gaussian-smoothing the guidance image [2] in 1994. The corresponding PDE can be written as follows:

$$u_t = -\mathrm{sgn}\big((u * K_\sigma)_{\eta\eta}\big)|\boldsymbol{\nabla} u| \tag{2.16}$$

where K_σ is a Gaussian with standard deviation σ. In Chapter 7, we investigate the Gaussian-smoothing of the guidance image further.

In 2003, Weickert proposed the Coherence Enhancing Shock Filter [32]. It uses the guidance image $v = (u * K_\sigma)_{ww}$ where $\boldsymbol{w}$ is the dominant eigenvector of the structure tensor. The speed function is the sgn function:

$$u_t = -\mathrm{sgn}\big((u * K_\sigma)_{ww}\big)|\boldsymbol{\nabla} u| \tag{2.17}$$

The behavior of this filter is explained in Chapter 7 in more detail.

In 2000, Schavemaker et al. [26] proposed a shock filter definition that uses non-flat morphology with paraboloids as structuring functions instead of the flat dilation and erosion operators. They use the second derivative in gradient direction in the guidance term. The corresponding evolution is given by

$$u_t = -\mathrm{sgn}(u_{\eta\eta})|\boldsymbol{\nabla} u|^2 \tag{2.18}$$

Chapter 3

Second Derivative Operators

In the last chapter, we introduced our general formulation of shock filtering. One important part of that is the second order derivative operator Lv, that steers the shock filter. For signals that are not just one dimensional, such as digital images, there are unlimited possibilities to create derivative operators if directional derivatives are used. We will analyze and compare shock filters based on their guidance term in the following chapters. In this chapter, we introduce the derivative operators that are used in the guidance term of the shock filters.

3.1 Laplacian

The Laplacian is one very successful and well-known operator in image processing. It is defined as the linear combination of second order directional derivatives in orthogonal directions:

$$\Delta u = u_{xx} + u_{yy} = u_{\eta\eta} + u_{\xi\xi} \qquad \boldsymbol{\eta} \perp \boldsymbol{\xi} \tag{3.1}$$

An abrupt change of the intensity of image values can be seen as an edge. Such a change of intensity corresponds to maxima of the first derivative and with that to zero crossing of the second derivative. Thus, the zero crossings of the Laplacian can be used to identify edge locations. For example, Marr and Hildreth use the Laplacian in their edge detection operator, the Laplacian of Gaussians (LoG) [20]. That means that the Laplacian carries important structural information. The importance of the information given by the zero crossings of the Laplacian has been examined by Elder in [9] and by Hummel and Moniot in [14].

We discretize the Laplacian by means of the central difference approximation of the second partial derivatives:

$$(\Delta u)_{i,j} = (u_{xx})_{i,j} + (u_{yy})_{i,j} = \frac{u_{i+1j} - 2u_{ij} + u_{i-1j}}{h_x^2} + \frac{u_{ij+1} - 2u_{ij} + u_{ij-1}}{h_y^2} \tag{3.2}$$

3.2 Gradient Direction

Similar to the Laplacian, derivatives in direction of the spatial gradient are often used as an edge indicator. The first derivative this direction equals the gradient magnitude, which can be used as an edge indicator. If the gradient magnitude is very large at a position, then it is a candidate for an edge. For that reason it is used in many applications. For example, the Canny Edge detector [6] uses the gradient magnitude for identifying edge candidates.

Since maximal values of the first derivative in gradient direction can correspond to edge locations, the zero crossings of the second derivative in gradient direction mark these potential maxima. So the second derivative in gradient direction acts as an edge indicator as well. Its potential for edge detection has been examined by Haralick in [13].

The second derivative in gradient direction can be defined by using the Hessian:

$$u_{\eta\eta} = \frac{\boldsymbol{\nabla} u^T \mathcal{H}(u) \boldsymbol{\nabla} u}{|\boldsymbol{\nabla} u|^2} = \frac{u_x^2 u_{xx} + 2u_{xy}u_x u_y + u_y^2 u_{yy}}{|\boldsymbol{\nabla} u|^2} \tag{3.3}$$

where $\boldsymbol{\eta} = \frac{\boldsymbol{\nabla} u}{|\boldsymbol{\nabla} u|}$. This definition is discretized by replacing all partial derivatives by their central difference approximation.

$$(u_{\eta\eta})_{i,j} = \frac{(u_x)_{i,j}^2 (u_{xx})_{i,j} + 2(u_{xy})_{i,j}(u_x)_{i,j}(u_y)_{i,j} + (u_y)_{i,j}^2 (u_{yy})_{i,j}}{(u_x)_{i,j}^2 + (u_y)_{i,j}^2} \tag{3.4}$$

We use this derivative in shock filtering later on. If only the sign of the derivative is relevant and not its magnitude, we drop the normalization, to avoid dividing by zero. To ease readablility, this operator will be abbreviated with L^{∇}.

3.3 Eigendirection of the Structure Tensor

The structure tensor is a matrix, that averages gradient information within a neighborhood specified by convolution with a Gaussian kernel K_ρ. Its eigenvectors point to the directions of the largest and smallest local contrast.

Like Weickert in [32], we refer to the eigenvector with the larger corresponding eigenvalue as the dominant eigenvector.

Much like the gradient, the dominant eigenvector typically points across edges, since this is usually the direction with the larger local contrast. Therefore, the zero crossings of the second directional derivative in direction of the dominant eigenvector of the structure tensor can be used as a fuzzy edge detector.

We define the second derivative in direction of the normalized dominant eigenvector $\boldsymbol{w} = (w_1, w_2)^T$ by means of the Hessian:

$$u_{ww} = \boldsymbol{w}^T \mathcal{H}(u) \boldsymbol{w} = w_1^2 u_{xx} + 2u_{xy} w_1 w_2 + w_2^2 u_{yy} \tag{3.5}$$

The second derivatives are approximated using central differences. For readability, we will abbreviate this operator with L_ρ^w in the following chapters. We use the implementation of the second derivative in direction of the dominant eigenvector of the structure tensor, that Joachim Weickert provided.

3.4 Comparison

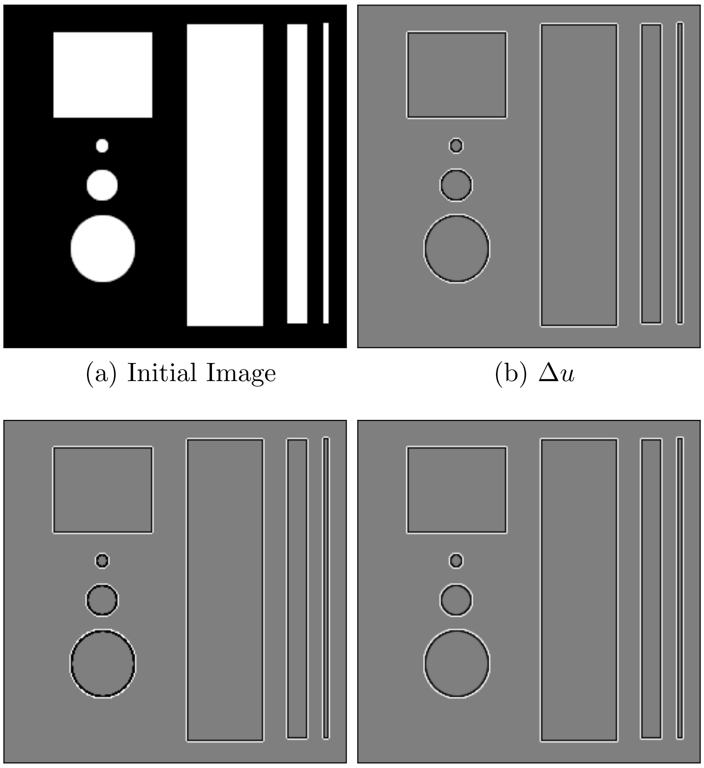
On important similarity of the three presented derivative operators is that they can be used in edge detection. For all three operators a change of the sign of the respective second derivative marks a possible edge location. Figure 3.1 shows the edge detection properties of the derivatives. Bright grey values correspond to a positive sign and dark ones to a negative sign. All three operators show a change of sign at the edges of the shapes depicted in the original image.

3.4.1 Differences between the Operators

Even though the different derivatives have similarities, they also have several differences. The following paragraphs explain some of the differences between the operators.

Conflicting Signs The Laplacian consists of two directional derivatives in perpendicular directions. Thus, if the signs of the directional derivatives are different, the Laplacian takes on the sign of the directional derivative with the larger magnitude. That can lead to a significant difference between the Laplacian an directional derivatives. For example, the Laplacian can be expressed by means of the gradient:

$$\Delta u = u_{\eta\eta} + u_{\xi\xi} \qquad \boldsymbol{\eta} || \boldsymbol{\nabla} u \text{ and } \boldsymbol{\xi} \perp \boldsymbol{\eta} \tag{3.6}$$



(a) Initial Image (b) Δu

(c) $L^{\nabla}u$ (d) $L^w_\rho u$, $\rho = 5$

Figure 3.1: Edge detection using second derivative operators. (a) Initial image. (b)-(d) Result of applying the different derivative operators. The result is shifted by 127.

1	24	25	25	25
1	24	25	25	25
1	13	15	15	25
1	24	25	25	25
1	24	25	25	25

(a) u

+	-	-	0	0
+	-	-	-	0
+	+	+	+	-
+	-	-	-	0
+	-	-	0	0

(b) $\mathrm{sgn}(\Delta u)$

+	-	-	0	0
+	-	-	-	0
+	-	-	+	-
+	-	-	-	0
+	-	-	0	0

(c) $\mathrm{sgn}(L^{\nabla})u$

+	-	-	0	0
+	-	-	-	-
+	-	-	+	-
+	-	-	-	-
+	-	-	0	0

(d) $\mathrm{sgn}(L^{w}_{\rho})u$, $\rho = 1$

Figure 3.2: Comparison of signs of the Laplacian and directional derivatives for an image that leads to conflicting signs of the Laplacian. (a) Original image u. (b) - (d) Signs of derivative operators.

Looking at this formulations shows, that the Laplacian can have a different sign from the second derivative in gradient direction if $|u_{\xi\xi}| > |u_{\eta\eta}|$. Of course, the same holds also for the derivative in direction of the dominant eigenvector of the structure tensor.
Figure 3.2 shows an example of this effect. In the original image, there is a large and abrupt change of intensity between the first and the second column, which could be interpreted as an edge. There is also a smaller change of intensity between the third row and the fourth and second row. While the derivatives in gradient direction and direction of the dominant eigenvector of the structure tensor have zero crossings between the first and second column, the Laplacian only shows a gap in this edge. This leads to a different structural interpretation of the image. Both directional derivatives seem to separate the image into two rectangular regions. The result produced by the Laplacian can be interpreted as three regions or two regions with a rather unclear edge.

There is one noteworthy problem that follows from conflicting signs of the Laplacian. If the signs of the partial derivatives are not equal, but their magnitudes are, then the derivatives cancel each other out and the Laplacian is zero, while the directional derivative, for example in gradient direction, are not.

0	0	0	0	0
0	0	0	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0

(a) u

0	0	0	0	0
+	+	+	0	0
-	-	-	+	0
0	0	-	+	0
0	0	-	+	0

(b) $\mathrm{sgn}(L^{\nabla}u)$

0	0	0	0	0
+	+	+	+	0
-	-	-	+	0
0	0	-	+	0
0	0	-	+	0

(c) $\mathrm{sgn}(L^{w}_{\rho}u)$, $\rho = 0.5$

Figure 3.3: Comparison of the signs of different derivative operators at corners. (a) Original image u. (b) - (d) Signs of derivative operators.

Corners The dominant eigenvector of the structure tensor carries information on the orientation of the local image structure. Therefore, for certain structures, the directional derivative in that direction behaves differently from the derivative in gradient direction. An example for these structures is given by corners.

Figure 3.3 illustrates this difference at a corner. For almost all pixels, the signs of the derivative operators are equal. However, the directional derivative in direction of the dominant eigenvector of the structure tensor has a positive sign at the top right pixel at the corner. The other derivative does not have that. This happens because in the discrete case the gradient only takes the directly adjacent pixel values into account, while the structure tensor considers the local image structure. The effect of that is that the zero crossings of $L^{w}_{\rho}u$ surround the object completely, while the zero crossings of $L^{\nabla}u$ do not do that.

Chapter 4

Shock Filtering with a Fixed Guidance Image

In Chapter 2, we purposely defined the class of shock filters relatively open. This allows us to use a guidance image, that is not equal to the evolving image or even connected to the initial image at all. In such a case, the guidance image does not evolve along with the initial image. So if no other evolution is applied to the guidance image, it is fixed. In this chapter, we explain how such a shock filter with a fixed guidance image behaves. First, we show some theoretical properties of the filter in the next section.

4.1 Properties

If we fix the guidance image and therefore also the guidance term, we can simplify the definition of the shock filter (Eq. (2.6)). We split the image domain Γ into three subsets by means of the sign of the guidance term:

$$\begin{aligned}
\Gamma &= \Gamma^{+} \cup \Gamma^{-} \cup \Gamma^{0} \\
\Gamma^{+} &= \{\boldsymbol{x} | Lv(\boldsymbol{x}) > 0\} \\
\Gamma^{-} &= \{\boldsymbol{x} | Lv(\boldsymbol{x}) < 0\} \\
\Gamma^{0} &= \{\boldsymbol{x} | Lv(\boldsymbol{x}) = 0\} \qquad (4.1)
\end{aligned}$$

Consider a connected region $R \subset \Gamma^{+}$ in the original domain where $Lv > 0$. In this region, the shock filter definition simplifies to the erosion equation:

$$u_t = -|s(Lv)||\boldsymbol{\nabla} u|, \qquad \text{on R} \qquad (4.2)$$

$$(4.3)$$

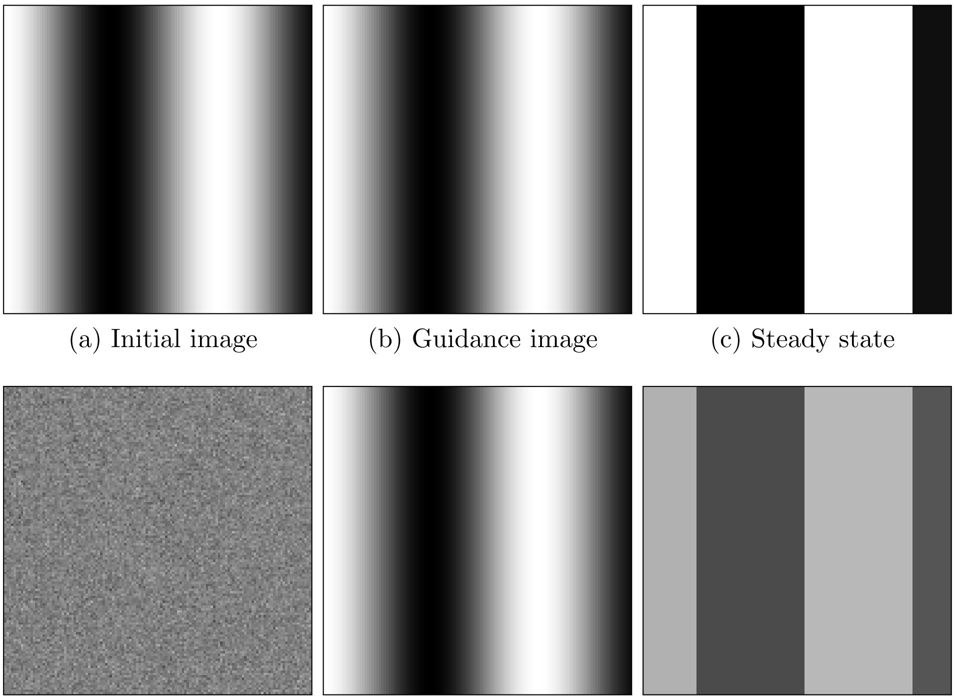
At the boundary of R the original values neighboring R are used. For regions in Γ^- and Γ^0 the reformulation is analogous. So if the guidance image is fixed, shock filtering locally simplifies to dilation and erosion. Thus, the shock filter also has many of the properties of dilation and erosion.
In 1993, Alvarez et al. [1] showed that dilation and erosion fulfill a minimum-maximum principle. The shock filter with a fixed guidance image also satisfies this minimum-maximum principle. If on all regions in Γ^- and Γ^+ the following holds:

$$\inf f \leq u(\boldsymbol{x}, t) \leq \sup f \tag{4.4}$$

then it also holds on Γ, since the values in Γ^0 are not changed.
In 1992, Jackway showed that dilation reduces the number of local maxima and erosion the number local minima. Both operations preserve the location of extrema. Likewise, shock filters preserve the location of extrema and reduce their number.
For $t \to \infty$, the evolution reaches a steady state that can be non-flat. On a connected region $R \subset \Gamma^+$, $u(\boldsymbol{x}, t)$ converges to the minimum of the original image f. On $R \subset \Gamma^-$ it converges to the maximal value of f in R. Values in Γ^0 are preserved.

4.2 Role of the Initialization

In a setting with a fixed guidance image, the guidance image can be chosen independently from the initial image. In that case, the shock filter changes the values of the evolving image u according to the structural information of v. The shock filter basically colors in the regions of the guidance image with the values of the initial image. The structure of the initial image does not effect the behavior of the shock filter in any way.
Figure 4.1 compares the result of the shock filter for different initial images. The guidance image is given by a cosine wave. Structurally, the resulting images look similar. In regions, where the cosine function is convex, the resulting image is dark and in concave regions, it is colored brighter. However, the corresponding dark and bright values of the initializations are different. For the resulting image that was initialized with the cosine wave, the bright and dark values are black and white. The noise initialization yields different grey values.



(a) Initial image (b) Guidance image (c) Steady state

(d) Initial image (e) Guidance image (f) Steady state

Figure 4.1: Comparison of result using different initial images. Top row: cosine wave as initial image and guidance image. Bottom row: noise initialization and cosine wave as the guidance image. (a) and (d) Initial images 128×128 pixels. (b) & (e) Guidance images. (c) and (f) Steady state of the shock filter using the Laplacian as the derivative operator. Images (a)-(c) and (e) are rescaled to the range of [0, 255] using affine rescaling.

4.3 Comparing Derivative Operators

The result of the shock filter strongly depends on the structure of the guidance image and therefore also on the second derivative operator that detects the structure. As we explained in Chapter 3, we mainly use the Laplacian and the second derivatives in direction of the gradient and the dominant eigenvector of the structure tensor. In the remainder of this section, we investigate the similarities and differences of the derivative operators for a fixed guidance term.

4.3.1 Equality in Strictly Convex/Concave Regions

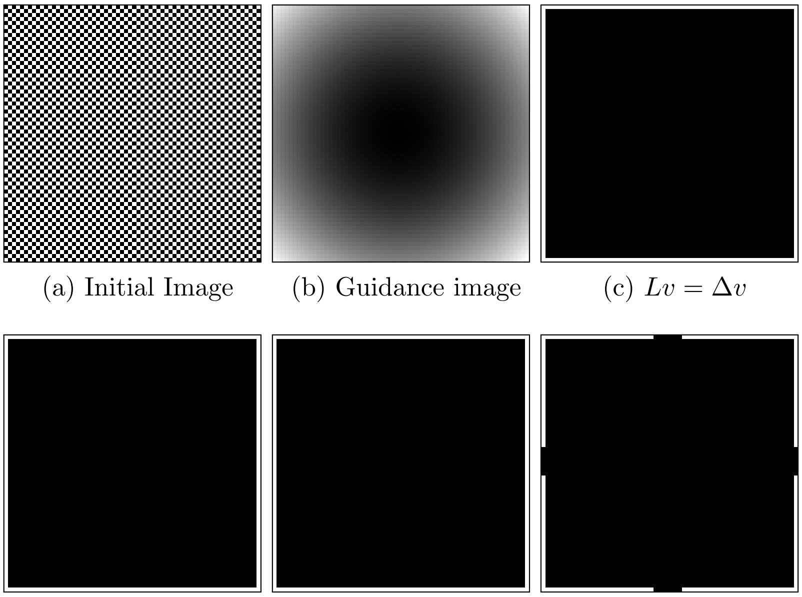
We can relate the behavior of shock filters to influence zones of minima and maxima, similar to the one dimensional case. In the influence zone of a local maximum, the Hessian is negative definite and in the influence zone of a local minimum it is positive definite. On an open and convex subsets of the influence zone of a maximum, the function is strictly concave and on an open and convex subset of the influence zone of a minimum, the function is strictly convex. A convex set of points on which the function is strictly (convex), is called strictly (convex) region in the following paragraphs. Analogously, we refer to such a set on which the function is (strictly) concave as a (strictly) concave region. In strictly convex/concave regions all shock filters with a fixed guidance term that use a second derivative operator of the following form behave equally:

$$Lv = \sum_i \partial_{p_i p_i} v \tag{4.5}$$

where $\boldsymbol{p_i}$ is a normalized vector.
We can show this by using the relation between the definiteness of the Hessian and the convexity/concavity of a function. On a convex, open set $R \subset \mathbb{R}^2$, a function is strictly convex if the Hessian is positive definite on that set. A matrix $\boldsymbol{A}$ is positive definite if for any direction $\boldsymbol{p} \neq \boldsymbol{0}$, $\boldsymbol{p}^T \boldsymbol{A} \boldsymbol{p} > 0$. Therefore, for an arbitrary, normalized direction $\boldsymbol{p_i} \neq \boldsymbol{0}$ the following holds:

$$0 < \boldsymbol{p_i}^T \mathcal{H}(v) \boldsymbol{p_i} = \partial_{p_i p_i} v \tag{4.6}$$

So all second directional derivatives in a strictly convex region must be positive. Thus, all second derivative operators of the form described in Eq. (4.5) are positive. A shock filter using such a derivative operator always performs erosion on a strictly convex region.
The strictly concave case is analogous. A matrix $\boldsymbol{A}$ is negative definite if



(a) Initial Image (b) Guidance image (c) $Lv = \Delta v$

(d) $Lv = L^{\nabla} v$ (e) $Lv = L^{w}_{\rho} v$ (f) $Lv = L^{w^{\perp}}_{\rho} v$

Figure 4.2: Comparison of shock filter with different derivative operators using a fixed, convex guidance image. (a) Initial image: checkerboard pattern. (b) Guidance image of the function $u(x, y) = x^2 + y^2$. (c)-(f): Steady state of shock filter with different second derivative operators. (c) Laplacian. (d) Derivative in gradient direction. (e) Derivative in direction of the dominant eigenvector of the structure tensor. (f) Second derivative in direction of the weak eigenvector of the structure tensor ($\boldsymbol{w} \perp \boldsymbol{w}^{\perp}$). The guidance image was rescaled to the range of [0, 255] using affine rescaling.

$\boldsymbol{p}^T \boldsymbol{A} \boldsymbol{p} < 0$ holds for any $\boldsymbol{p} \neq \boldsymbol{0}$. Let the function be strictly concave on the convex, open set R. Then on this set, the second directional derivative in an arbitrary direction $\boldsymbol{p_i} \neq 0$ must be positive

$$0 > \boldsymbol{p_i}^T \mathcal{H}(v) \boldsymbol{p_i} = \partial_{p_i p_i} v \tag{4.7}$$

So in strictly concave regions, derivative operators of the form from Eq. (4.5) are negative and shock filters using such an operator always perform dilation.

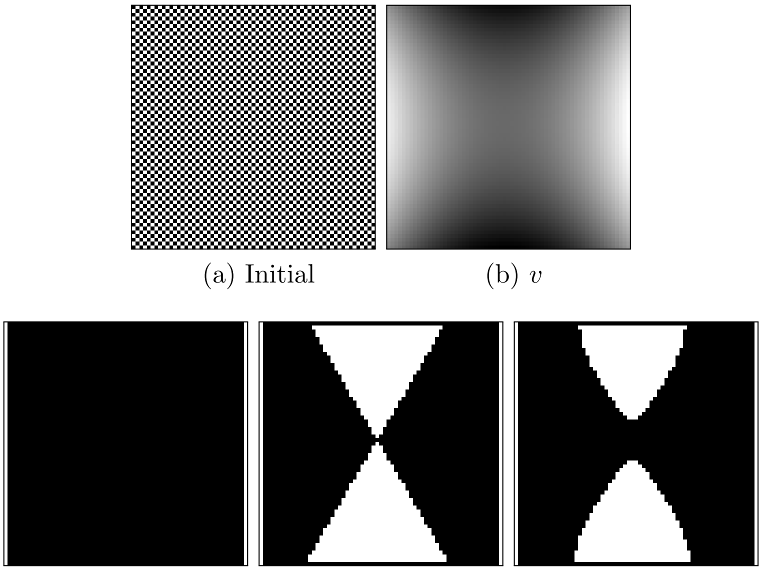
Figure 4.2 demonstrates the equality of shock filters with fixed guidance image in a strictly convex region. It shows the steady states of shock filtering a convex image using the Laplacian and the second derivative in direction of the gradient, the direction of the dominant eigenvector of the structure tensor and the direction of the other eigenvector of the structure tensor. All four resulting images consist of a black square surrounded by a white frame. Since the reflecting boundary conditions cause the pixels at the image border to lose their convexity, the different shock filters do not have to behave equally in these border pixels. Comparing the images 4.2e and 4.2f shows that outside of the strictly convex region, the second derivatives in directions of the eigenvectors of the structure tensor can behave differently, even though they behave equally in the strictly convex region.

Remark In regions that are not strictly convex or concave but convex or concave, the conclusion from above is not as strong. In such a case, there is at least one direction $\boldsymbol{p}$ for which $\partial_{pp}u = 0$. So in convex regions, shock filters never perform dilation and in concave region they never perform erosion. If you consider the example from Figure 4.1 again, you can see that the image is constant in y-direction. Thus the second derivative in y-direction is also 0 everywhere. Derivative operators that consider derivatives across edges are usually still positive in convex and negative in concave regions.

4.3.2 Differences

In regions that are neither convex nor concave, the derivative operators do not necessarily act equal. In the remainder of this thesis, we often refer to those regions as indefinite regions.
An example of shock filtering such an indefinite region is given in Figure 4.3. The function depicted in the guidance image is $v(x, y) = x^2 - 0.7y^2$. In the center of the image, the function has a saddle point. The slope in x and y direction is different, so the function has a directional preference. The Laplacian performs erosion almost everywhere, except for at the boundaries, since the Laplacian of v is positive everywhere. The results using the directional derivatives are similar, but not equal. The derivative in gradient direction results in two triangular bright regions. If you look closely, you can see that the dark shapes at the left and right of the triangles are connected. This connection is also created and even enhanced by the shock filter using the directional derivative in direction of the dominant eigenvector of the structure tensor.

The shock filter colors the regions defined by the guidance term in the brightest or darkest value of the corresponding region. That means that for a shock filter with a fixed guidance image, the differences between the behavior of the shock filter corresponds to the differences between the derivative operators themselves. In Figure 4.4 we reuse the example from Figure 3.2 from Chapter 3, in which we showed the effect of the Laplacian having conflicting signs. Here, the guidance image corresponds to the original image in the previous figure. The effect we explained earlier is also visible. On the left side of



(a) Initial (b) v

(c) $Lv = \Delta v$ (d) $Lv = L^{\nabla} v$ (e) $Lv = L^{w}_{\rho} v$, $\rho = 5$

Figure 4.3: Effect of shock filter using a fixed, neither convex nor concave guidance image. (a) Initial image: checkerboard pattern. (b) Guidance image of function $v(x, y) = x^2 - 0.7y^2$. (c)-(e): Steady state of shock filter with different second derivative operators. (c) Laplacian. (d) Derivative in gradient direction. (e) Derivative in direction of the dominant eigenvector of the structure tensor. The guidance image was rescaled to the range of [0, 255] using affine rescaling.

1	0	1	0	1
0	1	0	1	0
1	0	1	0	1
0	1	0	1	0
1	0	1	0	1

(a) Initial Image u

1	24	25	25	25
1	24	25	25	25
1	13	15	15	25
1	24	25	25	25
1	24	25	25	25

(b) Guidance Image v

0	1	1	0	1
0	1	1	1	0
0	0	0	0	1
0	1	1	1	0
0	1	1	0	1

(c) $Lv = (\Delta v)$

0	1	1	0	1
0	1	1	1	0
0	1	1	0	1
0	1	1	1	0
0	1	1	0	1

(d) $Lv = (L^{\nabla} v)$

0	1	1	0	1
0	1	1	1	1
0	1	1	0	1
0	1	1	1	1
0	1	1	0	1

(e) $Lv = (L^w_\rho v)$, $\rho = 1$

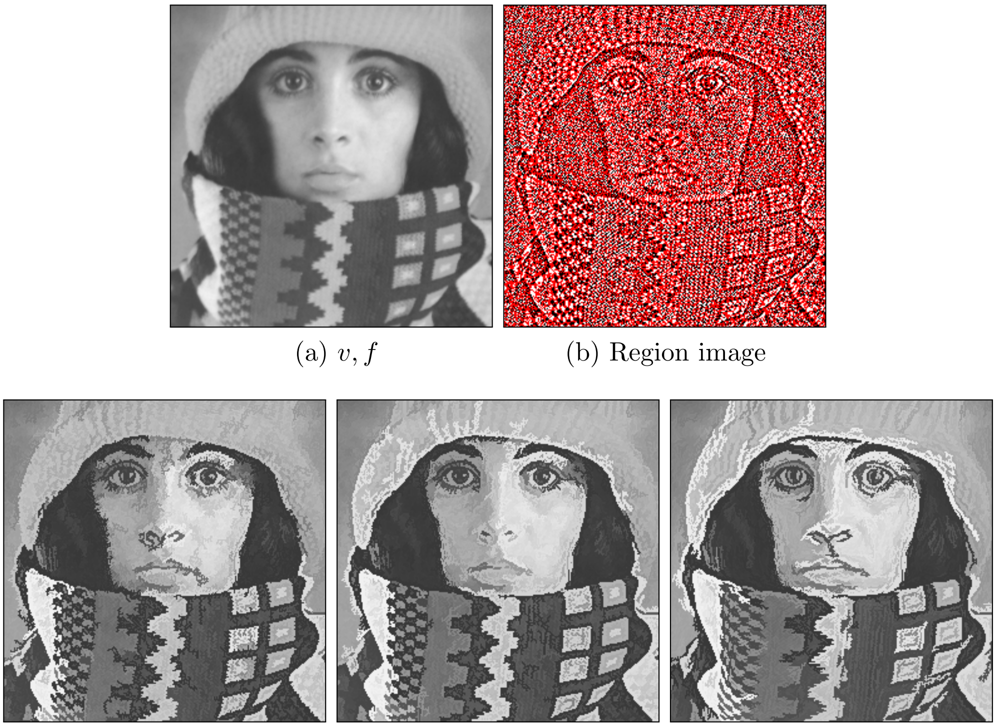
Figure 4.4: Steady states of different shock filters with a guidance image leading to conflicting signs of the Laplacian. (a) Original image: checkerboard pattern of size 5×5. (b) Guidance image. (c) - (e) Steady state of shock filters with different derivative operators.

the edge between the first and second row, all guidance terms have a positive sign and therefore, the filters all perform erosion. Thus, in the steady state, the values on the left side of that edge are 0. On the right side, $(L^{\nabla} v)$ and $(L^w_\rho v)$ have negative signs and therefore perform dilation. In the middle, of the second and third column, the Laplacian still performs erosion, since the conflicting signs lead to a positive sign. Therefore, the edge produced by the shock filter using the Laplacian is not as clear as the edge produced by the other two shock filter.

4.4 Natural Images

The properties of shock filters with regard to the convexity and concavity can be used to explain the behavior of shock filters with different derivative operators in natural images. Figure 4.5 shows the steady states of shock filtering the trui image using different derivative operators. The trui image is used as both the initial and the guidance image. Figure 4.5b visualizes strictly convex regions by coloring them black, strictly concave regions by coloring them white and red regions are neither strictly convex nor strictly concave. One can see that the image is made up of many small strictly convex/concave and indefinite regions. Therefore the results of the shock

filters are similar, but not equal. The edges in the result of the shock filter that uses the Laplacian are not as clean as the edges generated by the two directional derivatives. The result that uses the second derivative in gradient direction has cleaner edges. The result of the Laplacian and the gradient look relatively similar, but some regions are interpreted differently. For example the nostrils are colored dark by the Laplacian and bright by the derivative in gradient direction. The difference of both results to one generated using v_{ww} is larger overall. In this image, the directional preference of structures plays a big role. This derivative operator is used in the Coherence Enhancing Shock Filter by Weickert [32]. Therefore, it is not surprising that using this derivative operator seems to enhance or elongate low-like structures, even though no Gaussian smoothing is involved.
There is one more aspect in all three images that we need to point out. All three image seem segmented according to their respective derivative operator. But as can be seen looking at the image in Figure 4.5b, the are many regions even if only the strict convexity is considered. Since the shock filter colors regions, this can lead to a over-segmentation.



(a) v, f

(b) Region image

(c) $Lv = \Delta v$

(d) $Lv = L^{\nabla} v$

(e) $Lv = L^{w}_{\rho} v$, $\rho = 5$

Figure 4.5: Effect of shock filters using a fixed, natural guidance image. (a) Guidance image and initial image. (b) False color representation of strict convexity/concavity of the image. Strictly convex regions are black, strictly concave regions white and red regions are not strictly convex/concave. (c)-(e): Steady state of shock filter with different second derivative operators.

Chapter 5

Evolving Guidance Image

An interesting case arises if the guidance image is set to the evolving image $v = u$. This leads to the following image evolution:

$$u_t = -\operatorname{sgn}(Lu)|\boldsymbol{\nabla} u| \tag{5.1}$$

Replacing Lu by the second derivative in gradient direction leads to the shock filter definition of Osher and Rudin from [21]. Unfortunately, there are currently no well-posedness results for this PDE available [35]. Therefore, we only work with the discrete definition of the shock filter and the derivative operators.

5.1 Strictly Convex and Concave Regions

In the previous chapter, we showed that all shock filters with a fixed guidance term behave equally in strictly convex and concave regions. In the case of an evolving guidance image, it is not quite as simple, since the evolution affects the derivatives of the image. Nevertheless, the behavior in strictly convex/concave regions is still interesting.

First, consider the strictly concave case. Here a shock filter step always applies dilation, since all directional derivatives are negative. Therefore, we need to investigate the influence of dilation on the concavity of the function. We assume that concave and convex regions consist of more than one pixel.

We first consider an inner pixel (i, j), for which all neighbors are convex and u is not maximal in (i, j).

If $(u_{pp})^k_{i,j} > 0$, then image at the time $k + 1$ is then given by

$$u^{k+1}_{i,j} = u^k_{i,j} - \tau\Bigg(\max\left\{0, \frac{u^k_{i,j} - u^k_{i+1,j}}{h_x}, \frac{u^k_{i,j} - u^k_{i-1,j}}{h_x}\right\}^2$$

$$+ \max\Big\{0, \frac{u^k_{i,j} - u^k_{i,j+1}}{h_y}, \frac{u^k_{i,j} - u^k_{i,j-1}}{h_y}\Big\}^2\Big)^{\frac{1}{2}} \tag{5.2}$$

Without loss of generality, we assume that in both x- and y-direction the backward difference is smaller and that $|u^k_{i,j} - u^k_{i+1,j}| \leq |u^k_{i,j} - u^k_{i-1,j}|$. Therefore, the following holds:

$$\begin{aligned}
u^{k+1}_{i,j} &= u^k_{i,j} - \tau\sqrt{\Big(\frac{u^k_{i,j} - u^k_{i+1,j}}{h_x}\Big)^2 + \Big(u^k_{i,j} - \frac{u^k_{i,j+1}}{h_y}\Big)^2} \\
&\geq u^k_{i,j} - 2\tau\frac{u^k_{i,j} - u^k_{i+1,j}}{h_x}) \\
&= u^k_{i,j}\Big(1 - 2\frac{\tau}{h_x}\Big) + \frac{2\tau}{h_x}u^k_{i+1,j} && (5.3) \\
& && (5.4)
\end{aligned}$$

For the second directional derivative in the normalized and arbitrary direction $\boldsymbol{p}$ at time level $k+1$, the following holds:

$$\begin{aligned}
(u_{pp})^{k+1}_{i,j} = &\Big(p_1^2\frac{u^{k+1}_{i+1,j} - 2u^{k+1}_{i,j} + u^{k+1}_{i-1,j}}{h_x^2} + p_2^2\frac{u^{k+1}_{i,j+1} - 2u^{k+1}_{i,j} + u^{k+1}_{i,j-1}}{h_y^2} \\
&+ 2p_1p_2\frac{u^{k+1}_{i+1,j+1} + u^{k+1}_{i-1,j-1} - u^{k+1}_{i+1,j-1} - u^{k+1}_{i-1,j+1}}{4h_xh_y}\Big) \\
&\geq (1 - 2\frac{\tau}{h_x})\Big(p_1^2\frac{u^k_{i+1,j} - 2u^k_{i,j} + u^k_{i-1j}}{h_x^2} + p_2^2\frac{u^k_{i,j+1} - 2u^k_{i,j} + u^k_{i,j-1}}{h_y^2} \\
&+ 2p_1p_2\frac{u^k_{i+1,j+1} + u^k_{i-1,j-1} - u^k_{i+1,j-1} - u^k_{i-1,j+1}}{4h_xh_y}\Big) \\
&+ 2\frac{\tau}{h_x}\Big(p_1^2\frac{u^k_{i+2,j} - 2u^k_{i+1,j} + u^k_{i,j}}{h_x^2} + p_2^2\frac{u^k_{i+1,j+1} - 2u^k_{i+1,j} + u^k_{i+1,j-1}}{h_y^2} \\
&+ 2p_1p_2\frac{u^k_{i+2j+1} + u^k_{i,j-1} - u^k_{i+2,j-1} - u^k_{i,j+1}}{4h_xh_y}\Big) && Eq.(5.3) \\
&= (1 - 2\frac{\tau}{h_x})(u_{pp})^k_{i,j} + 2\frac{\tau}{h_x}(u_{pp})^k_{i+1,j} && (5.5)
\end{aligned}$$

So for $\tau \leq \frac{h_x}{2}$

$$(u_{pp})^{k+1}_{i,j} \geq 0 \tag{5.6}$$

Therefore, τ should be chosen such that $\tau \leq \min h_x, h_y$ and for $h_x = h_y = 1$, $\tau \leq 0.5$.

The argumentation for the strictly concave case is very similar. It can be shown, that dilation preserves concavity just as erosion preserves convexity. For better readability, the corresponding calculation is moved to the appendix 11.

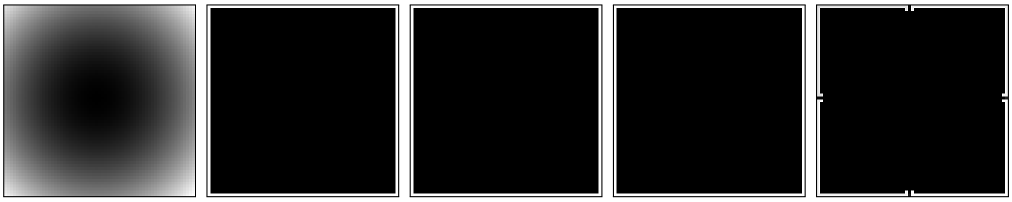
Let now (i, j) be the location of a local maximum. Dilation preserves the location of maxima [35]. Therefore, $u_{i,j}^{k+1}$ is still a local maximum and u is still concave in (i, j). Analogously, if there is a minimum in (i, j) the shock filter does not change the value in that pixel and u remains convex in that pixel since the values around the minimum can not become smaller than the minimum.

For pixels on the border of convex/concave regions, such a general statement can not be made. Thus, shock filters do not have to behave equally in border pixels.
In summary, we can say that inner pixels of a strictly concave region increase towards the maximum in the region. In border pixels, the behavior may be different. However, the regions tend to evolve towards a flat region with the value of the maximum. Analogously, strictly convex regions tend to become flat regions with the value of the local minimum.
Figure 5.1 shows that by the example of an initially strictly convex image, with non-convex border pixels. All three derivative operators flatten the strictly convex region, such that it takes on the value of the minimum. Here one can also see the white border which was also present the fixed case. Looking at the image in Figure 5.1e shows, that shock filters in pixels at the border of a convex region may behave differently. At the left, right, top and bottom border of the image white pixels are visible inside the originally convex region.

Note that the findings from this sections are true for regions that are initially strictly convex/concave. If the region is only convex or concave then some derivatives are 0. In that case, a derivative operator that only uses such a direction will also be zero. If a derivative operator is used that is typically non zero in convex/concave regions, such as the Laplacian or the second derivative in gradient direction, then the region tends to become a flat region during the evolution.

5.2 Non-convex/concave Regions

Like in the case of a fixed guidance image, the behavior of the shock filter depends on the derivative operator in regions that are neither convex nor concave. However, other than in the fixed case, the guidance term may change its sign now.



(a) u, f (b) $Lu = \Delta u$ (c) $Lu = L^{\nabla} u$ (d) $Lu = L^{w}_{\rho} u$ (e) $Lu = L^{\nabla^{\perp}} u$

Figure 5.1: Shock filters with an evolving guidance image applied to a convex image. (a) Initial image of the function $u(x, y) = x^2 + y^2$. (b)-(d): Steady state of shock filter with different second derivative operators. (b) Laplacian. (c) Derivative in gradient direction. (d) Derivative in direction of the dominant eigenvector of the structure tensor $\rho = 2$. (e) Derivative in direction opposite of the gradient $T = 125$. The images are rescaled to the range of [0, 255] using affine rescaling.

For a second directional derivative in an arbitrary direction $\boldsymbol{p} = (p_1 p_2)^T$ at time level $k + 1$ that yields:

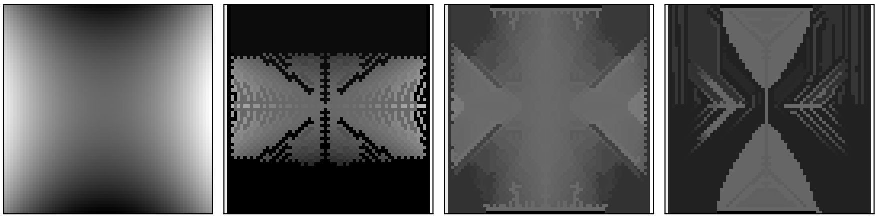
$$
\begin{aligned}
(u_{pp})^{k+1}_{i,j} &= \frac{1}{|\boldsymbol{p}|^2} \Big(p_1^2 \frac{u^{k+1}_{i+1,j} - 2u^{k+1}_{i,j} + u^{k+1}_{i-1,j}}{h_x^2} + p_2^2 \frac{u^{k+1}_{i,j+1} - 2u^{k+1}_{i,j} + u^{k+1}_{i,j-1}}{h_y^2} \\
&+ 2p_1 p_2 \frac{u^{k+1}_{i+1,j+1} + u^{k+1}_{i-1,j-1} - u^{k+1}_{i+1,j-1} - u^{k+1}_{i-1,j+1}}{h_x h_y} \Big) \\
&= (u_{pp})^{k}_{i,j} + \tau (\delta_{pp})^{k}_{i,j}
\end{aligned} \tag{5.7}
$$

Thus, the sign of $(u_{pp})^{k}_{i,j}$ changes if the second derivative in the same direction of the change of the image δ has the opposite sign and a larger magnitude. Note that $(\delta_{pp})^{k}_{i,j}$ is scaled by the time step size τ. So a larger time step size makes a change of sign more likely.

The derivative operators that we have considered so far are not simply directional derivatives in a fixed direction. Therefore, there are some special aspects that need to be mentioned. The Laplacian is composed of directional derivatives in both x and y direction. So to determine if there is a change of sign of the Laplacian, Equation 5.7 has to be changed as follows:

$$
(\Delta u)^{k+1}_{i,j} = (\Delta u)^{k}_{i,j} + \tau (\Delta \delta)^{k}_{i,j} \tag{5.8}
$$

So the sign of the Laplacian changes, if the Laplacian of the change of u has an opposite sign and a larger absolute value.
The directional derivatives in direction of the gradient and of the dominant eigenvector of the structure tensor may also change their direction.

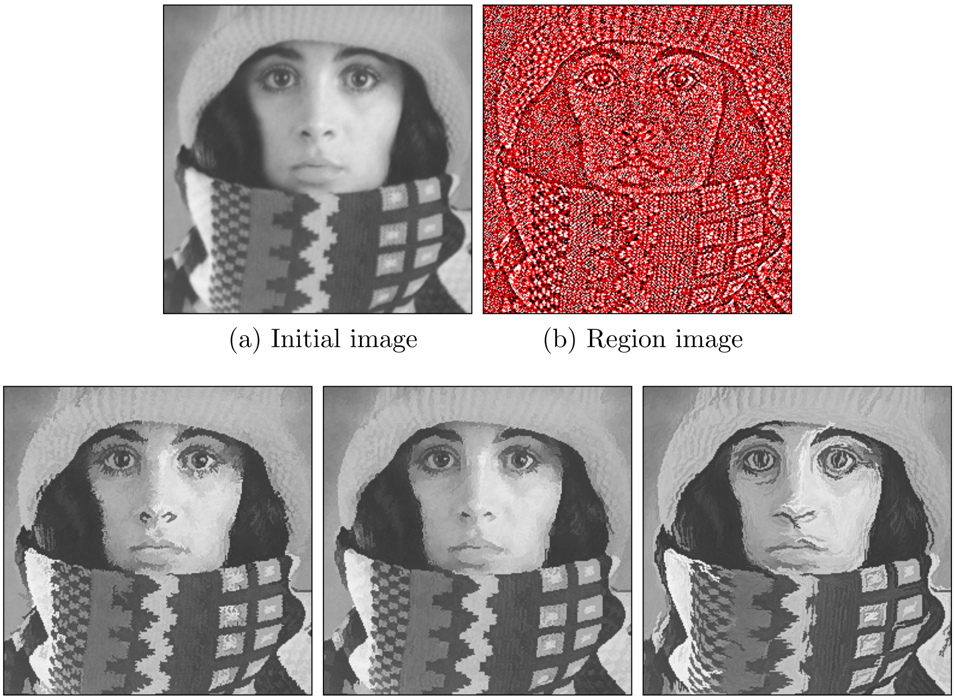


(a) u, f (b) $Lu = \Delta u$ (c) $Lu = L^{\nabla} u$ (d) $Lu = L^{w}_{\rho} u$

Figure 5.2: Shock filters with evolving guidance image applied to an indefinite region. (a) Initial image of $u(x, y) = x^2 - 0.7y^2$. (b)-(d): Steady states of shock filter with different second derivative operators. $\tau = 0.5$ for all images. For (d) $\rho = 5$. The images are rescaled to the interval [0, 255] using affine rescaling.

5.3 Natural Images

Natural images consist of many small convex, concave and indefinite regions. Due to potentially different behavior at borders, not all shock filters behave completely equally in strictly convex/concave regions. Nevertheless, they behave similarly. Thus, using the three different derivative operators in shock filtering yields similar, but not equal results. Figure 5.3 shows the steady states of shock filtering the trui image with an evolving guidance image. Similar to the fixed guidance image, you can see the properties of the derivative operators in these images.



(a) Initial image (b) Region image

(c) $Lu = \Delta u$ (d) $Lu = L^{\nabla} u$ (e) $Lu = L^{w}_{\rho} u$, $\rho = 5$

Figure 5.3: Shock filtering natural image with evolving guidance image. (a) Original image. (b) Strictly convex(black)/concave(white) of the initial image. Red regions are neither strictly convex nor strictly concave. (c)-(d)Steady states of the shock filters. $\tau = 0.5$.

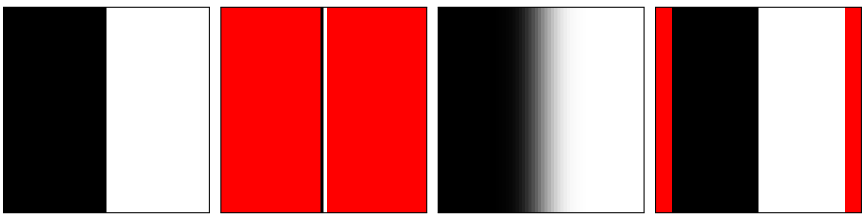
Chapter 6

Deblurring with Shock Filtering

Originally, Kramer and Bruckner proposed shock filters as a tool for image deblurring and noticed that deblurring drawn lines or characters, works especially well [16]. In this chapter, we show that shock filters can be used for deblurring and explain why they work especially well for cartoon-like images.

6.1 How Blur Effects Edges

Blur is a type of image degradation that has many possible causes including defocussing, motion during image creation or atmospheric disturbances [33]. Mathematically, blur is modeled as a convolution. The convolution kernel determines the type of blur. For example Gaussian convolution can be used to model atmospheric disturbances.



(a) (b) (c) (d)

Figure 6.1: Effect of Gaussian blur. (a) Initial image. (b) Convex and concave regions of (a). (c) Gaussian blur with standard deviation $\sigma = 5$. (d) Convex and concave regions of (c). Convex regions are black, concave regions white and indefinite regions are red.

Edges in particular are strongly affected by blur. Consider the example in Figure 6.1. An edge like the one given in this example has a line of convex pixels (not strictly convex) left of the edge and a concave one on the right of the edge. If Gaussian convolution is applied then not only the grey values at pixels are averaged, but by the differentiation property of convolution also the derivatives:

$$\begin{aligned} \partial_v u_\sigma &= \partial_v (u * K) = (\partial_v u) * K \\ \partial_{vv} u_\sigma &= \partial_{vv} (u * K) = (\partial_{vv} u) * K \end{aligned} \tag{6.1}$$

where K is the convolution kernel. Therefore, the initially convex and concave strips right and left of an edge are enlarged. Figure 6.1 also depicts that for a Gaussian as a convolution kernel.

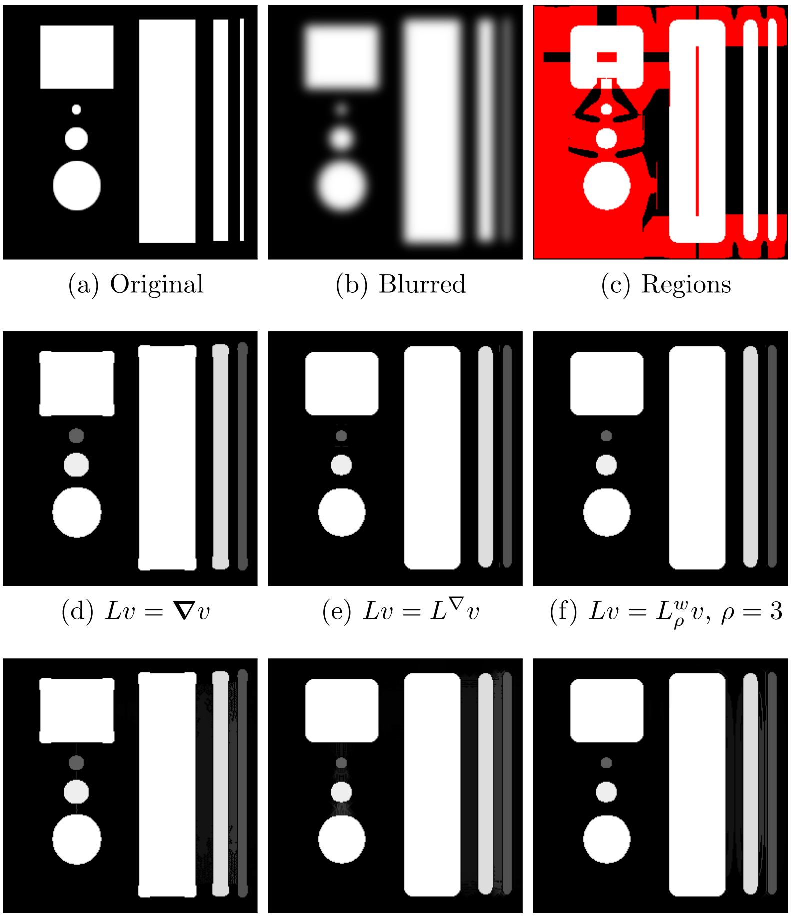
6.2 Deblurring Cartoon-Like Images

Cartoon-like images are images with clear edges and flat regions. Their edges and regions are comparable to the edge from Figure 6.1. If blur is applied to such an image, larger convex and concave regions are created. As we already explained, in regions that are only convex/concave shock filter with derivative operators that consider the direction across the edge apply dilation in concave regions and erosion in convex regions. Therefore, a shock filter using one of the three derivative operators we have been using so far, sharpens the blurred edges until the steady state.

Figure 6.2 compares the deblurring of an initially binary image by different shock filters. The blur is relatively strong which causes the convex regions that should surround the objects to blend into each other and lose the convexity in some parts of the image. The concave regions inside the objects are mostly preserved. Therefore, the results of the different shock filters are all relatively similar. All shock filters manage to paint the concave regions inside the object in a bright grey value. Outside of the concave regions the shock filters do not always behave equally. For example the Laplacian creates only slightly rounded corners at the rectangles, while the derivatives in direction of $\boldsymbol{\nabla} u$ and $\boldsymbol{w}$ yield really rounded corners. Interestingly, the shock filters with a fixed guidance image yield a better result than those that use evolving guidance images. The shock filters with fixed guidance images create a clear separation between the different objects, such that the result strongly resembles the initial image. The separation of the objects created using evolving guidance images is not as clear. Even in the steady state, some objects like the disks or the two leftmost rectangles are still connected with a dark colored region, which is not the case if a fixed guidance image is used.

6.3 Deblurring Natural Images

Deblurring natural images has additional difficulties. Natural images typically do not have flat regions. Usually, they are filled with grey value gradients. However, applying a shock filter until it reaches a steady state will also flatten regions, that are not supposed to be flattened. In Figure 6.3 this problem is highlighted. For the small stopping time of $T = 0.75$ the image is already sharpened, but not segmented extremely strong. For a larger evolution time, the images look more segmented and the segments become flat.

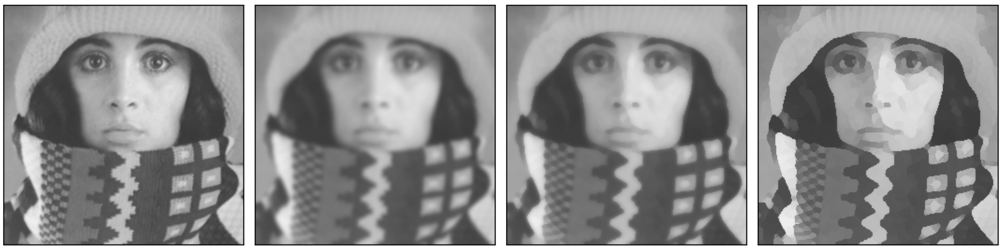


(a) Original (b) Blurred (c) Regions

(d) $Lv = \boldsymbol{\nabla} v$ (e) $Lv = L^{\nabla} v$ (f) $Lv = L^{w}_{\rho} v$, $\rho = 3$

(g) $Lv = \Delta u$ (h) $Lv = L^{\nabla} u$ (i) $Lv = L^{w}_{\rho} u$, $\rho = 3$

Figure 6.2: Deblurring a blurred, cartoon-like image with shock filtering. Top row: (a) Initial image. (b) Gaussian-blurred image with standard deviation $\sigma = 5$, initial image and guidance image for the shock filter with fixed guidance image. (c) Convex and concave regions of (b). Convex regions are black, concave regions white and indefinite regions are red. Middle row: Steady states of shock filters with fixed guidance image applied to (b). The guidance image is the blurred image. Bottom row: Steady states of shock filters with evolving guidance image.



(a) Original (b) Blurred (c) $T = 0.75$ (d) $T = 5$

Figure 6.3: Deblurring a blurred, natural image with shock filtering. (a) Initial image. (b) Gaussian-blurred image with standard deviation $\sigma = 2$. (c)-(d) Shock filter with guidance term $Lu = L^{\nabla}u$ at different stopping times.

Chapter 7

Gaussian-Smoothing of the Guidance Image

Derivative computation is generally very noise sensitive. Alvarez and Mazorra suggest that applying Gaussian-smoothing to the guidance image may solve the noise sensitivity problem [2]. Also, the Gaussian-smoothing may enlarge or fuse different regions. This might solve the problem of over-segmentation. Therefore, we investigate the effect of smoothing the guidance image in this chapter.

Gaussian convolution is a very popular tool for regularizing images. Therefore, it has also been used in shock filtering. In 1994, Alvarez and Mazorra [2] first proposed presmoothing the guidance image using Gaussian convolution for their shock filter. In their definition, the guidance image equals the evolving image. They use the second derivative in gradient direction for the derivative operator. With that their shock filtering process for the evolving image u can be expressed as follows:

$$u_t = -s(L^{\nabla} u_\sigma)|\boldsymbol{\nabla} u| \tag{7.1}$$

where $u_\sigma = u * K_\sigma$ for the Gaussian K_σ. Gaussian presmoothing is also used by Weickert's Coherence Enhancing Shock Filter [32]. Instead of the second flow line derivative, this shock filter uses $L^w_\rho u_\sigma$. Both shock filters show interesting, yet very different results, which are presented later in this section.

Convolution of an image with a Gaussian of standard deviation σ is equivalent to applying homogeneous diffusion until the stopping time of $T = \frac{\sigma^2}{2}$ [35]. We use this equivalence to formulate this modification of the guidance image as an image evolution itself. If the shock filter uses a fixed guidance image v, then the input of the evolution is always the initial guidance image

v^0:

$$\begin{aligned} v_{i,j}^k &= \Delta v_{i,j}^0 \\ u_{i,j}^{k+1} &= -\mathrm{sgn}(Lv_{i,j}^k)(|\boldsymbol{\nabla} u|)_{i,j}^k \end{aligned} \tag{7.2}$$

In the the case of $v = u$, the filter is applied to the evolving image u:

$$\begin{aligned} v_{i,j}^k &= \Delta u_{i,j}^k \\ u_{i,j}^{k+1} &= -\mathrm{sgn}(Lv_{i,j}^k)(|\boldsymbol{\nabla} u|)_{i,j}^k \end{aligned} \tag{7.3}$$

The Gaussian convolution is implemented using a convolution in the image domain with a sampled Gaussian that is truncated and normalized. The implementation was given to us by Joachim Weickert.

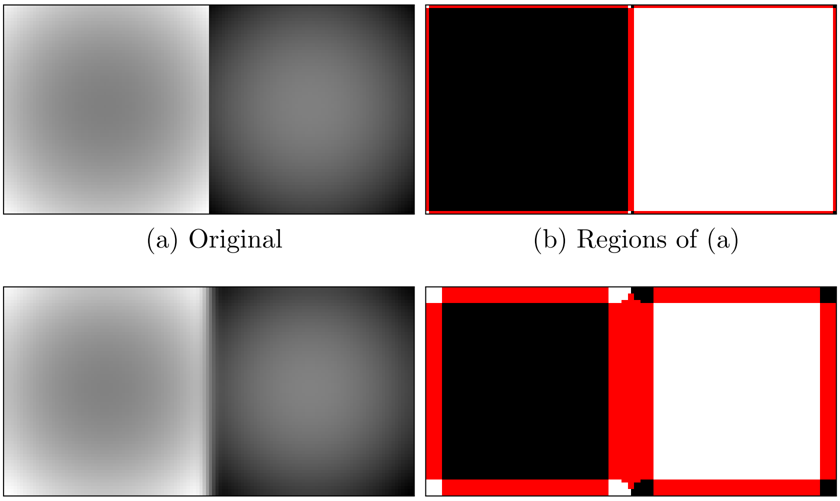
7.1 Effect on (strict) Convexity and Concavity

In the previous chapters, the relevance of convexity/concavity of image regions was highlighted several times. Thus, to understand the effect of presmoothing the guidance image, we need to investigate the effect that the smoothing process has on convex/concave regions.

Convolution averages values of an image within a window. For Gaussian convolution that window is defined by the Gaussian kernel. By the differentiation property of convolution, it also convolves the derivatives of that image within that window:

$$\begin{aligned} u_\sigma &= u * K_\sigma \\ \partial_v u_\sigma &= \partial_v (u * K_\sigma) = (\partial_v u) * K_\sigma \\ \partial_{vv} u_\sigma &= \partial_{vv} (u * K_\sigma) = (\partial_{vv} u) * K_\sigma \end{aligned} \tag{7.4}$$

This averaging can lead to a change of the sign of second derivatives. Imagine a strictly convex region located next to a strictly concave region. If Gaussian convolution is applied, then the sign of the second derivatives of the border pixels are influenced by derivatives with an opposite sign. Thus, they can change their sign. So the strictly convex and strictly concave regions shrink. Figure 7.1 depicts such a case. The initial image depicts a strictly convex paraboloid next to a strictly concave one. Initially, the image is strictly convex on the left side, strictly concave on the right and there is a non-convex/concave border between the regions. There are strictly concave pixels above and below the corners of the convex region. For the right side these



(a) Original (b) Regions of (a)

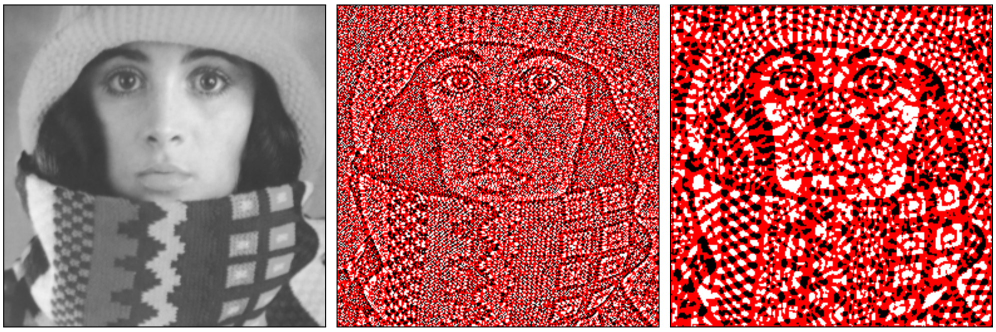
(c) Smoothed $\sigma = 2$ (d) Regions of (c)

Figure 7.1: Change of strictly convex/concave regions caused by Gaussian convolution. (a) Original image of two functions: Left $x^2 + y^2$, right $-x^2 - y^2$. (b) Regions of the original image that are strictly convex or concave or neither. (c) Smoothed original image. (d)Regions of the smoothed image that are strictly convex or concave or neither. Convex regions are marked black, concave regions white and regions that are neither red. The original and smoothed image were rescaled using affine rescaling.

pixels are strictly convex. After applying Gaussian convolution to the image, these regions at the corners and the non-convex/concave region are larger, while the large strictly convex/concave regions are shrunken. If the size of the kernel was large enough, the convex and concave region could even vanish completely after the convolution.

This is only one example of how Gaussian convolution can effect the convexity and concavity of an image. There are many more ways in which Gaussian convolution can influence second derivatives. For example it has been shown that Gaussian convolution can create new extrema [18, 39]. Thus, it can even create new convex and concave regions.

In summary, smoothing an image with a Gaussian can create, shrink, enlarge and erase strictly convex/concave regions. Of course, this has a large effect on the convexity and concavity of natural images. Figure 7.2 shows an example of that. Before smoothing, the image consists of many small strictly convex/concave regions and regions that are neither strictly convex nor strictly concave. After applying Gaussian-smoothing, there are less segments that are much larger.



(a) Original (b) Without smoothing (c) With smoothing

Figure 7.2: Change of strictly convex/concave regions of a natural image through Gaussian convolution. Strictly convex regions are marked black, strictly concave regions white and regions that are not strictly convex/concave red. (a) Original image. (b) Regions without smoothing. 45.3% of the pixels are colored red. (c) Regions after Gaussian-smoothing with $\sigma = 1.5$. 54.6% of the pixels are colored red.

Furthermore, the number of indefinite pixels increases with the convolution. Without smoothing, 45.3% of the pixels are neither strictly convex nor strictly concave. After applying Gaussian-smoothing, 54.6% of the pixels are colored red. However, Gaussian-convolution does not necessarily increase the number of pixels that are neither strictly convex nor strictly concave. Imagine an image with shows a black pixel in the middle of a white background. Here all pixels can be considered indefinite, except for the neighbors of the black pixel, in which the image is convex. Applying Gaussian-smoothing with a sufficiently small standard deviation to the image enlarges the convex region. So the number of indefinite pixels decreases. Using a sufficiently large standard deviation would cause the black pixel to vanish and with that also the strictly convex pixels.

7.2 Shock Filter with Smoothed Guidance Image

Gaussian convolution has a large effect on image derivatives and with that also on convexity and concavity. So it is no surprise that presmoothing the guidance image in shock filtering effects the result of the shock filter. We again differentiate between shock filtering with a fixed and an evolving guidance image.

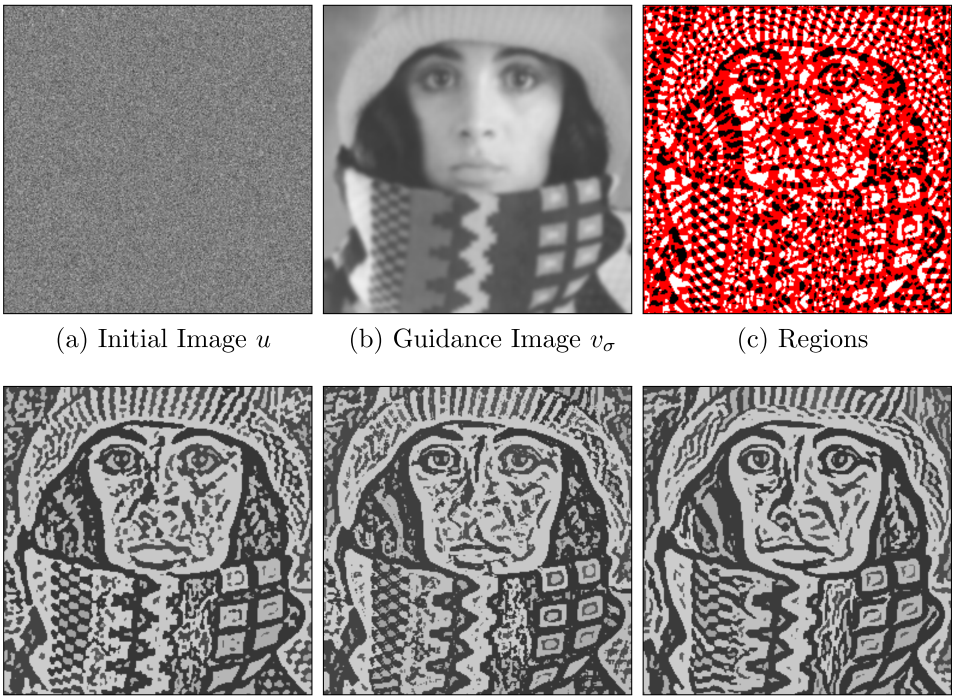
7.2.1 Fixed Guidance Image

If the guidance image is fixed, the presmoothing takes place before the shock filtering process. Therefore, the shock filtering process itself is not effected by the Gaussian convolution. Like before, all shock filters with derivative operators of the form $Lu = \sum_i u_{p_i}$ for any direction $\boldsymbol{p_i}$ behave equally inside strictly convex/concave regions. In non-convex/concave regions they do not necessarily behave similarly. Since Gaussian convolution enlarges these non-convex/concave regions and some of the strictly convex/concave regions, the guidance image has less regions and both the similarities between derivative operators and their differences are visible more clearly.

Figure 7.3 shows the results of shock filtering a noise image using a guidance image that is smoothed using Gaussian convolution with different derivative operators. In strictly convex/concave regions, the images look similar. In the other regions, the differences such as the directional preference of $L^w_\rho v$ are clearly visible. For example at the checkerboard pattern on the scarf, there are strictly convex and strictly concave regions that are separated a by non-convex/concave regions. Here the results using the Laplacian and the second flow line derivative look similar, while the shock filter using $L^w_\rho v$ almost destroys the pattern due to the directional preference. In many regions, one can also see that the edges created by the shock filters using the $L^\nabla v$ and $L^w_\rho v$ are cleaner than those created using the Laplacian. These differences were somewhat visible without the presmoothing if one considers Figure 4.5 from Chapter 4, but the convolution accentuates these effects, since there are more and larger indefinite regions.

7.2.2 Evolving Guidance Image

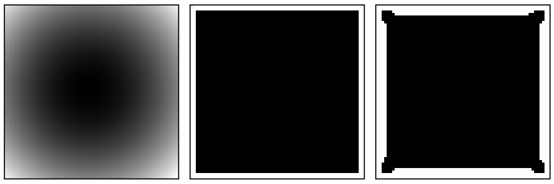
In the special case $u = v$ the convolution is applied in every iteration. Thus, the averaging of the derivatives used in the guidance term happens in every shock filter step. Here, the equivalence of homogeneous diffusion to Gaussian convolution is important. With increasing time, diffusion propagates information throughout the image domain. So repeatedly applying Gaussian convolution also propagates information from one point in the image to another point. In shock filtering, the convolution is only applied to the guidance image, so the convolution does not directly propagate values. However, the dilation/erosion part paints the different regions according to the sign of the second derivative operator. So the transport of the information by the convolution may be carried over to the evolving image through the dilation/erosion. Since in shock filtering the averaging can be understood as an averaging of the image derivatives, the behavior of the propagation



(a) Initial Image u (b) Guidance Image v_σ (c) Regions

(d) $Lv = \Delta v$ (e) $Lv = L^{\nabla} v$ (f) $Lv = L^{w}_{\rho} v$, $\rho = 5$

Figure 7.3: Steady states of differentshock filters with a Gaussian-smoothed, fixed guidance image.. Presmoothing using Gaussian convolution with a kernel with $\sigma = 1.5$.



(a) (b) (c)

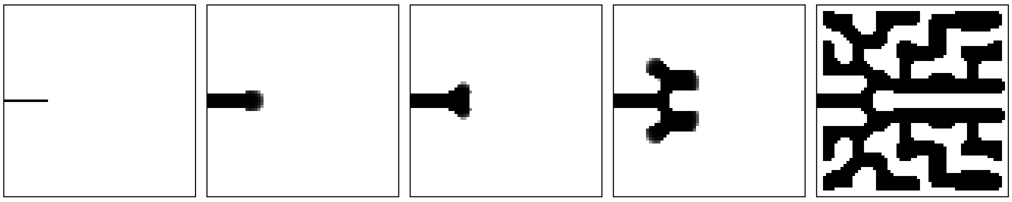
Figure 7.4: Steady states of shock filters with $\tau = 0.7$ using the guidance term $Lu = \Delta u_\sigma$ for a strictly convex initial image. (a) Initial image of the function $u(x, y) = x^2 + y^2$. (b) $\sigma = 0.5$. (c) $\sigma = 2$. All images were rescaled to the range of [0, 255] using affine rescaling.

heavily depends on which derivatives are used. In the following paragraphs, we present the propagation properties of shock filters for different derivative operators in different experiments.

Laplacian

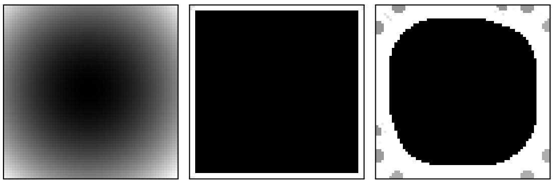
The Laplacian is a symmetric derivative operator that consists of derivatives in two perpendicular directions. Thus, the propagation caused by the convolution also acts symmetrically and in two perpendicular directions. Figure 7.4 shows how the convolution influences the behavior of a shock filter with evolving guidance image, if the Laplacian is used as the derivative operator. The initial image is a convex function. For a very small narrow convolution kernel, the result looks like the result without the convolution. If the kernel is large enough, the results of the shock filter looks different. The white edge, that a shock filter without presmoothing creates, is thickened. Moreover, there is a gap in the border at the edges. This is caused by the averaging of information from the for example the upper and left border.

Figure 7.5 demonstrates the evolution of a line-like object under shock filtering with an evolving smoothed guidance term that used the Laplacian. The filter first thickens the line until it its thickness corresponds to the standard deviation of the Gaussian. Simultaneously, a blob-like structure forms at the end of the line. After 20 iterations, the structure that formed at the end of the line spreads out in two directions forming new structures. This continues with each iteration. At 50 iterations, more of these structures have developed already. In the steady state, the image is filled with structures that developed from this procedure.



(a) Initial (b) 10 (c) 20 (d) 50 (e) Steady state

Figure 7.5: Evolving an image with a line like structure using a shock filter with the guidance term $Lu = \Delta u_\sigma$. $\sigma = 2$ and $\tau = 0.5$. (a) Initial image. (b) - (d) result after increasing number of iterations. (e) Steady state.



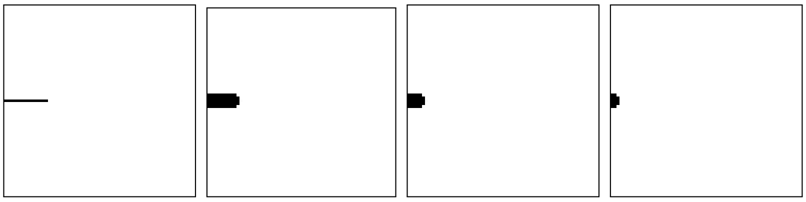
(a) (b) (c)

Figure 7.6: Steady states of shock filters with $\tau = 0.7$ using the guidance term $Lu = L^\nabla u_\sigma$ for a strictly convex initial image. (a) Initial image of the function $u(x, y) = x^2 + y^2$. (b) $\sigma = 0.5$. (c) $\sigma = 2$. All images were rescaled to the range of [0, 255] using affine rescaling.

Flow-Line Derivative

Using the second derivative in gradient direction as the derivative operator yields the guidance term $L^\nabla u_\sigma$. A shock filter using this operator was proposed by Alvarez and Mazorra [2]. They extended the shock filter of Osher and Rudin [21] with Gaussian smoothing to make the shock filter more robust against noise.
We performed the same experiments for the second derivative in gradient direction as for the Laplacian. Figure 7.6 depicts the result of applying a shock filter using $L^\nabla u_\sigma$ as the derivative operator to a convex image. For a small convolution kernel, the result looks like it does without the presmoothing for this derivative operator as well. For a larger convolution kernel the shock filter changed the structure of the image. The steady state consists of a white border surrounding a black region, but the structure is slightly round.

Figure 7.7 shows again the evolution of the line under the shock filter using $L^\nabla u_\sigma$. Like before, the line is thickened to the size of σ. However, other than for the Laplacian, the line shrinks towards the image border here.



(a) Initial (b) 10 (c) 20 (d) Steady state

Figure 7.7: Evolution of an image with a line like structure using a shock filter with the guidance term $Lu = L^{\nabla} u_{\sigma}$. $\sigma = 2$ and $\tau = 0.5$. (a) Initial image. (b) - (c) result after increasing number of iterations. (d) Steady state.

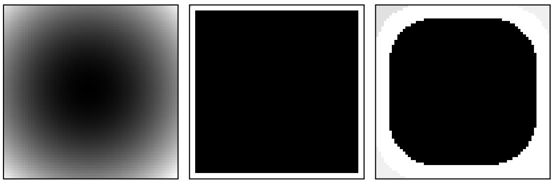
Initially, the point of the largest change of grey values is at the rightmost pixel of the line. The convolution moves that point to the left. Therefore, on the left of that point the intensity change increases until that point. Finally, on the right of that point the change of intensity decreases again. Thus, the second derivative in gradient direction must be positive on the left and negative on the right. So dilation is applied on the right of that new point, which causes the black line to become shorter with each iteration. In the steady state, only the round end of the line created by the convolution is visible. It is not removed completely by the shock filter as before, since the edge is very close to the image border and can not be moved any further.
In summary, it seems like shock filters that use the guidance term $L^{\nabla} u_{\sigma}$ tend to round larger structures and shrinks structures of a scale that is related to the size of the Gaussian that is used in the convolution.

Coherence Enhancing Shock Filter

Using the second derivative in direction of the dominant eigenvector of the structure tensor as the second derivative operator and an evolving guidance image that is presmoothed with Gaussian convolution yields the Coherence Enhancing Shock Filter that was proposed by Weickert in 2003 [32].

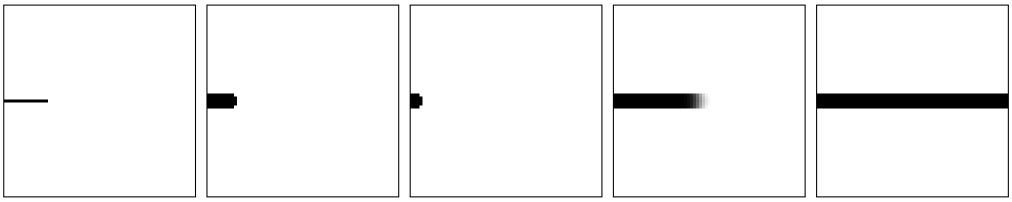
Figure 7.8 shows the result of the experiment using a convex initial image if Coherence Enhancing Shock Filtering is used. For a very small Gaussian kernel, the filter behaves like it did in the other two cases. If the kernel is sufficiently large, the shock filter enlarges the border around the dark center. The dark center keeps a squarish shape but its corners are rounded. This rounding at the corners comes from the round shape of the Gaussian.

Figure 7.9 demonstrates the evolution of the line under the Coherence Enhancing Shock Filter. The results use different values for the integration scale. For a small integration scale, the direction of dominant eigenvector of



(a) (b) (c)

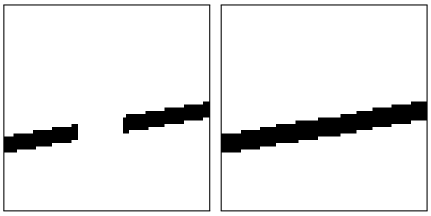
Figure 7.8: Steady states of shock filters with $\tau = 0.7$ using the guidance term $Lu = L^w_\rho u_\sigma$, $\rho = 5$, for a strictly convex initial image. a) Initial image of the function $u(x, y) = x^2 + y^2$. b) $\sigma = 0.5$. c) $\sigma = 2$. All images were rescaled to the range of [0, 255] using affine rescaling.



(a) Initial (b) $\rho = 1$, 20 iterations (c) $\rho = 1$, Steady state (d) $\rho = 5$, 20 iterations (e) $\rho = 5$, Steady state

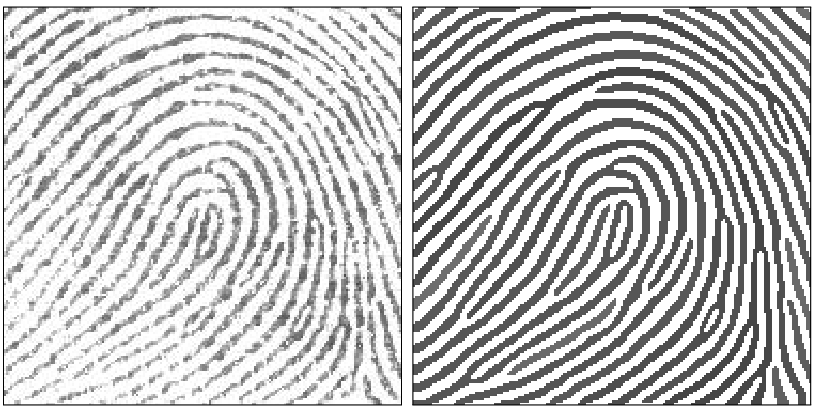
Figure 7.9: Evolution of a line like structure using a shock filter with the guidance term $Lu = L^w_\rho u_\sigma$. (a) Initial image 65×65 pixels. (b) - (c) Result after increasing number of iterations for $\rho = 1, \sigma = 2$ and $\tau = 0.7$. (d) - (e) Result after increasing number of iterations for $\rho = 5, \sigma = 2$ and $\tau = 0.7$.

the structure tensor is relatively close to the direction of the gradient. So if a small integration scale is chosen, the result using the Coherence Enhancing Shock Filter is similar to the result using the shock filter of Alvarez and Mazorra: the line is thickened and shrinks towards the image border. Using a larger integration scale yields a different result. Here the line is thickened and elongated towards the opposite image border. Figure 7.10 explores the ability of the Coherence Enhancing Shock Filter to connect line fragments. The initial image shows a line with a gap. The shock filter manages to connect both line fragments, while keeping a matching slope.
In [32], Weickert shows that the Coherence Enhancing Shock Filter can be used to enhance coherence structures in general by using it to enhance a fingerprint. Figure 7.11 shows their example. The initial image is a fingerprint with partially disconnected and unclean lines. In the steady state of the shock filter the lines are connected and slightly thicker. The generated edges are sharp.



(a) Initial (b) Steady state

Figure 7.10: Connecting lines with Coherence Enhancing Shock Filtering.(a) Initial image 64×64 pixels. (b) Steady state of the Coherence Enhancing Shock Filter applied to (a), $\rho = 4$, $\sigma = 2$ and $\tau = 0.7$ was used.



(a) Initial (b) Steady state

Figure 7.11: Enhancing a fingerprint with Coherence Enhancing Shock Filtering. (a) Initial image 186×186 pixels from [32]. (b) Steady state, $\rho = 5$, $\sigma = 1.5$ and $\tau = 0.7$ was used.

Summary

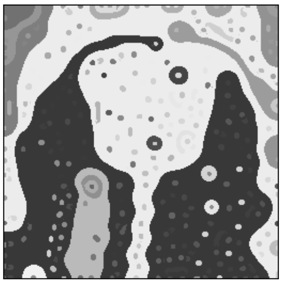
We can use these findings to explain the behavior of different shock filters that use an evolving, smoothed guidance image. Figure 7.12 compares the effect of the different shock filters if they are applied to the trui test image. One can observe the effects that we described in the precious paragraphs in the resulting images. The results produced by the shock filter using the Laplacian show that pattern of linear structures being enlarged and split. A larger smoothing scale produces larger structures. If the second derivative in gradient direction is used, the filter produces roundish segments and removes smaller scale structures. Using the coherence enhancing filter leads to the enhancement of coherent structures of a scale that is defined be the size of the Gaussian. For a larger Gaussian, the formed structures have a larger scale and smaller scale details vanish.

In general, applying Gaussian-smoothing to an evolving guidance image leads to the generation of structures of scale of the size of the Gaussian. Smaller scale details vanish. This can lead to interesting effects such as the enhancement of coherent structures. However, it does not necessarily have a regularizing effect, even for a small smoothing scale. Comparing the results using $\sigma = 0.5$ from Figure 7.12 to the result of Figure 5.3 from Chapter 5 7, one may argue that the results using the Gaussian-smoothing look less smooth. To empathize this point, we applied the three shock filters to a noisy image in Figure 7.13. Since the image is extremely noisy, we used a rather larger smoothing scale. The three resulting images are not noisy anymore, however they do not really look like the the original image. One can clearly see the characteristics of the different shock filters. The Laplacian creates the pattern of splitting lines. The second derivative in gradient direction creates somehow roundish structures. The large structure at the top even slightly resembles the initial triangle.The Coherence Enhancing Shock Filter constructs coherent structures, that were not in the image initially. One can imagine this behavior of the shock filters trying to make sense of the noise and therefore creating structures according to their characteristics using the noise, which leads to results that may not even remotely resemble the original image.

(a) $Lu = \Delta u_\sigma$,
$\sigma = 0.5$

(b) $Lu = \Delta u_\sigma$,
$\sigma = 2$

(c) $Lu = L^\nabla u_\sigma$,
$\sigma = 0.5$ $T = 3500$

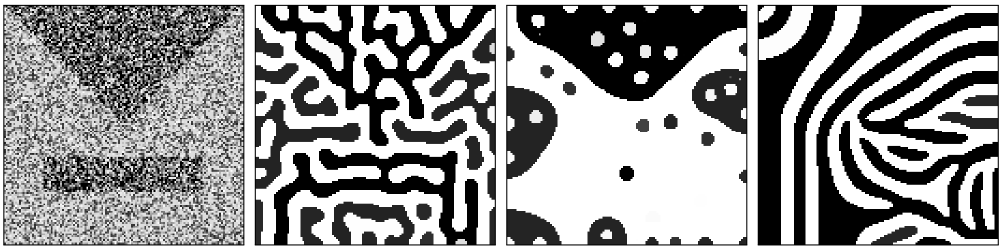


(d) $Lu = L^\nabla u_\sigma$,
$\sigma = 2$

(e) $Lu = L^w_\rho u_\sigma$,
$\sigma = 0.5$, $\rho = 5$

(f) $Lu = L^w_\rho u_\sigma$,
$\sigma = 2$, $\rho = 5$

Figure 7.12: Result of applying different shock filters with Gaussian-smoothed, evolving guidance image to a natural image (trui). All images, other than (c), are the steady state of the evolution. $\tau = 0.7$ was used for all images.



(a) Initial image. (b) $Lu = \Delta u_\sigma$ (c) $Lu = L^\nabla u_\sigma$ (d) $Lu = L^w_\rho u_\sigma$, $\rho = 5$

Figure 7.13: Applying shock filters with Gaussian-smoothed, evolving guidance image to a noisy image. (a) Initial image. (b) - (d) Steady states of different shock filters. $\sigma = 3$ and $\tau = 0.7$.

Chapter 8

Shock-Diffusion

In the previous chapter, we examined the combination of diffusion and shock filtering that uses Gaussian-smoothing in the guidance term. Unfortunately, this does not necessarily lead to a regularized shock filter. Another way of combining the two filters is coupling their PDEs. Such a coupling can be realized by using a weighted linear combination of a shock term $\mathcal{S}(u)$ and a diffusion term $\mathcal{D}(u)$:

$$u_t = \omega_D \mathcal{D}(u) + \omega_S \mathcal{S}(u) \tag{8.1}$$

where ω_D, ω_S are weights. We will be referring to this type of filter as shock-diffusion.

Such a linear combination of diffusion and shock term can actually be linked to an existing filter. In [15], Kornprobst et al. explain that nonlinear isotropic diffusion can be written in terms of such a combination. For convenience, we provide the computation here.
Nonlinear Isotropic Diffusion can be rewritten as follows:

$$\begin{aligned} u_t &= \operatorname{div}(g(|\boldsymbol{\nabla} u|^2)\boldsymbol{\nabla} u) \\ &= g(|\boldsymbol{\nabla} u|^2)\Delta u + 2g'(|\boldsymbol{\nabla} u|^2)|\boldsymbol{\nabla} u| u_{\eta\eta} |\boldsymbol{\nabla} u| \end{aligned} \tag{8.2}$$

The full computation can be found in Appendix Eq. 11.6.
Diffusivities typically have the form $\frac{1}{f(s^s)}$, where $f(s^2) > 0$. Therefore its derivative is negative and has the form:

$$g'(s^2) = -\frac{f'(s^2)}{f^2(s^2)} \tag{8.3}$$

Equation 8.2 can be fitted to the form of Equation 8.1, using the shock term $\mathcal{S}(u) = -\operatorname{sgn}(u_{\eta\eta})|\boldsymbol{\nabla} u|$, the diffusion term Δu and the weights $\omega_D =$

$g(|\boldsymbol{\nabla} u|^2)$, $\omega_S = 2\frac{f'(s^2)}{f^2(s^2)} u_{\eta\eta} |\boldsymbol{\nabla} u|$:

$$u_t = g(|\boldsymbol{\nabla} u|^2)\Delta u + 2\frac{f'(s^2)}{f^2(s^2)} u_{\eta\eta} |\boldsymbol{\nabla} u|(-\operatorname{sgn}(u_{\eta\eta})|\boldsymbol{\nabla} u|) \tag{8.4}$$

This connection to nonlinear isotropic diffusion makes the idea of a shock-diffusion filter seem reasonable. The combination of homogeneous diffusion and shock filtering has not really been investigated thoroughly thus far. Therefore, we investigate this combination in this chapter. First, we introduce some of the noteworthy related work, that investigates the combination of shock filtering and smoothing operators, apart from diffusion.

8.1 Related Work

Over the years, several linear combinations of smoothing and shock filter have been proposed [2, 15, 12, 11, 3, 38]. In this section, we briefly introduce the two shock-smoothing combinations that are the basis for all later proposed definitions.

The first combination of shock filtering with a smoothing operation was published by Alvarez and Mazorra in 1994 [2]. They propose to use a convex combination of Mean Curvature Motion and shock filtering. The shock term uses an evolving guidance image that is smoothed with Gaussian convolution. The derivative operator is the second derivative operator in direction of the gradient. Their proposed equation looks as follows:

$$u_t = C u_{\xi\xi} - \beta(L^{\nabla} u_\sigma, |\boldsymbol{\nabla} u_\sigma|)|\boldsymbol{\nabla} u| \tag{8.5}$$

where $\xi \perp \boldsymbol{\nabla} u$ and $C > 0$ is a constant that determines the weighting of both operators.

The definition of Kornprobst et al. [15] uses a linear combination of a shock term for image enhancement at edges, a smoothing term for denoising and a similarity term to force the convergence to a non-trivial steady state. The smoothing term uses smoothing in direction of the image gradient and the opposite direction. The direction of the gradient is weighted using a function $h_\alpha(|\boldsymbol{\nabla} u_\sigma|) = 1$ if $|\boldsymbol{\nabla} u_\sigma| < \alpha$ and 0 otherwise. Therefore, the process does not smooth across edges for large gradient magnitudes. In this definition, the same shock term is used as in the definition of Alvarez and Mazorra[2], but the shock term is weighted using $1 - h_\alpha(|\boldsymbol{\nabla} u_\sigma|)$. The equation is given by

$$u_t = \underbrace{c_f(u - f)}_{similarity} + \underbrace{c_d(h_\alpha(|\boldsymbol{\nabla} u_\zeta|) u_{\eta\eta} + u_{\xi\xi})}_{smoothing}$$

$$\underbrace{-c_s(1-h_\alpha(|\boldsymbol{\nabla} u_\zeta|))\mathrm{sgn}(L^{\nabla} u_\sigma)|\boldsymbol{\nabla} u|)}_{shock} \tag{8.6}$$

where c_f, c_d, c_s are constants, $\boldsymbol{\eta}||\boldsymbol{\nabla} u$ and $\boldsymbol{\eta}\perp\boldsymbol{\xi}$.

8.2 Model

Most linear combinations of smoothing and shock filtering use smoothing in direction of the flow lines and some additionally use smoothing in direction of the gradient. However, we use homogeneous diffusion for the smoothing term. For the shock term, we use an evolving guidance image. For the weighting of both terms, we use a $\omega_D = \omega(|\boldsymbol{\nabla} u_\zeta|^2)$ and $\omega_S = 1 - \omega(|\boldsymbol{\nabla} u_\zeta|^2)$, where $u_\zeta = K_\sigma * u$ for the Gaussian K_ζ with standard deviation ζ. This leads to the following formulation:

$$\begin{aligned} u_t &= \omega(|\boldsymbol{\nabla} u_\zeta|^2)\Delta u - (1-\omega(|\boldsymbol{\nabla} u_\zeta|^2))\mathrm{sgn}(Lu)|\boldsymbol{\nabla} u| && \text{on } \Gamma \times (0,\infty) \\ u_0 &= f && \text{on } \Gamma \times (0,\infty) \\ \partial_n u &= 0 && \text{on } \partial\Gamma \times (0,\infty) \end{aligned} \tag{8.7}$$

For the weighting function $0 \le \omega \le 1$ must hold. It should be chosen such that it is decreasing in the (squared) gradient magnitude.

A class of functions that satisfies these properties are diffusivities. In diffusion filtering, they are used to adapt the strength of the diffusion to the image structures.

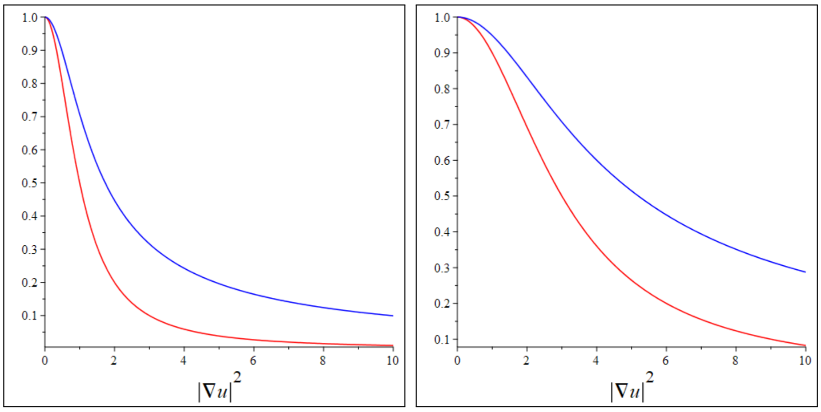
There are several different diffusivities. We use the Charbonnier diffusivity [8]

$$g_{CH}(s^2) = \frac{1}{\sqrt{1+\frac{s^2}{\lambda}}} \tag{8.8}$$

and the Perona-Malik diffusivity [22]

$$g_{PM}(s^2) = \frac{1}{1+\frac{s^2}{\lambda}} \tag{8.9}$$

The contrast parameter λ determines how large the influence of the gradient magnitude on the function is. A larger λ lessens the influence of the gradient magnitude. This has an effect on how fast the function gets close to zero. Figure 8.1 show the influence of the contrast parameter. For $\lambda = 1$ the functions decrease relatively fast, which leads to a small diffusivity for smaller values of the gradient magnitude. For $\lambda = 3$, the function does not decrease as drastically. Thus, for the same gradient magnitude the value of the diffusivity is larger. In general the Perona-Malik diffusivity decreases faster than the Charbonnier diffusivity.



(a) $\lambda = 1$ (b) $\lambda = 3$

Figure 8.1: Influence of the contrast parameter on the diffusivities. Red: Perona-Malik diffusivity. Blue: Charbonnier diffusivity.

8.2.1 Discretization

The shock term in Equation 8.7 is discretized as described in Chapter 2. The diffusion term is discretized using an explicit scheme with central differences:

$$\mathcal{D}_{i,j}^k = \left(\frac{u_{i+1,j}^k - 2u_{i,j}^k + u_{i-1,j}^k}{h_x^2} + \frac{u_{i,j+1}^k - 2u_{i,j}^k + u_{i,j-1}^k}{h_y^2}\right) \tag{8.10}$$

The explicit scheme for the discretization of diffusion is stable with respect to a minimum-maximum principle for $\tau \leq \frac{1}{\frac{2}{h_x^2}+\frac{2}{h_y^2}}$ [35].

The fully discrete scheme is given by

$$u_{i,j}^{k+1} = \tau(g_{i,j}^k \mathcal{D}_{i,j}^k + (1 - g_{i,j}^k)\mathcal{S}_{i,j}^k) + u_{i,j}^k \tag{8.11}$$

The shock term $S_{i,j}^k$ is discretized as described in Chapter 2.

Like the diffusion scheme, this scheme satisfies a discrete minimum-maximum principle for $\tau \leq \frac{1}{\frac{2}{h_x^2}+\frac{2}{h_y^2}}$.

At time level k, the following holds:

$$\begin{aligned} u_{i,j}^{k+1} &= \tau(g_{i,j}^k \mathcal{D}_{i,j}^k + (1 - g_{i,j}^k)\mathcal{S}_{i,j}^k) + u_{i,j}^k \\ &\leq \tau(g_{i,j}^k \max\{\mathcal{D}_{i,j}^k, \mathcal{S}_{i,j}^k\} + (1 - g_{i,j}^k)\max\{\mathcal{D}_{i,j}^k, \mathcal{S}_{i,j}^k\}) + u_{i,j}^k \\ &= u_{i,j}^k + \tau \begin{cases} g_{i,j}^k \mathcal{D}_{i,j}^k + (1 - g_{i,j}^k)\mathcal{D}_{i,j}^k, & \mathcal{D}_{i,j}^k > \mathcal{S}_{i,j}^k \\ g_{i,j}^k \mathcal{S}_{i,j}^k + (1 - g_{i,j}^k)\mathcal{S}_{i,j}^k, & \text{else} \end{cases} \end{aligned}$$

$$= \begin{cases} u_{i,j}^k + \tau \mathcal{D}_{i,j}^k, & \mathcal{D}_{i,j}^k > \mathcal{S}_{i,j}^k \\ u_{i,j}^k + \tau \mathcal{S}_{i,j}^k, & \text{else} \end{cases} \tag{8.12}$$

The first case is equivalent to the discretization of the diffusion and the second case to the discretization of the shock filter. Since both terms fulfill the minimum-maximum principle for $\tau \leq \frac{1}{\frac{2}{h_x^2}+\frac{2}{h_y^2}}$

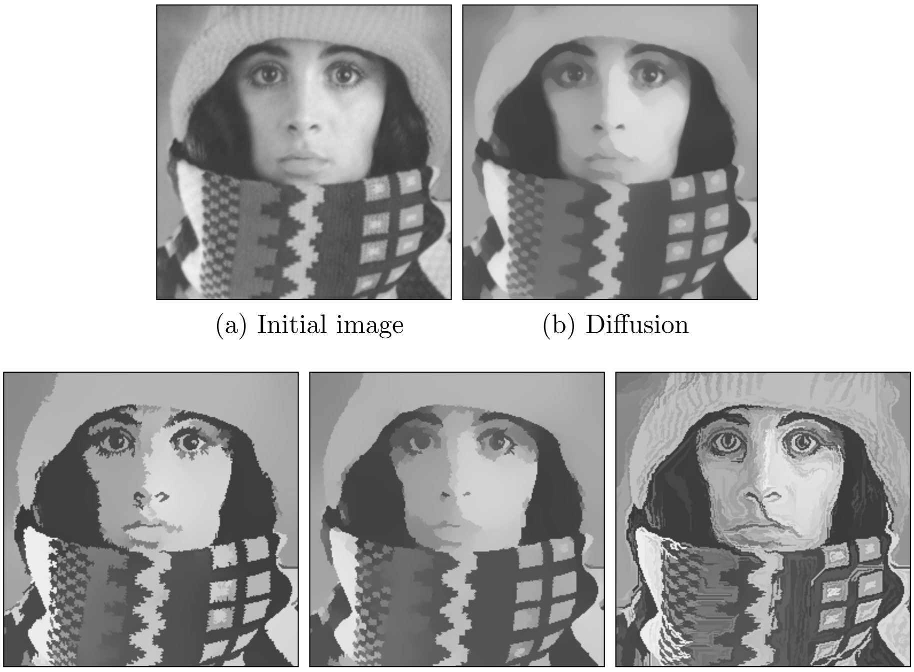
$$u_{i,j}^{k+1} = = \begin{cases} u_{i,j}^k + \tau \mathcal{D}_{i,j}^k, & \mathcal{D}_{i,j}^k > \mathcal{S}_{i,j}^k \\ u_{i,j}^k + \tau \mathcal{S}_{i,j}^k, & \text{else} \end{cases} \leq \max_{n,m \in \Gamma} \{f_{n,m}\} \tag{8.13}$$

Analogously, you can show that the evolution does not fall below the minimum.

$$\begin{aligned} u_{i,j}^{k+1} &= \tau(g_{i,j}^k \mathcal{D}_{i,j}^k + (1 - g_{i,j}^k)\mathcal{S}_{i,j}^k) + u_{i,j}^k \\ &\geq \tau(g_{i,j}^k \min\{\mathcal{D}_{i,j}^k, \mathcal{S}_{i,j}^k\} + (1 - g_{i,j}^k) \min\{\mathcal{D}_{i,j}^k, \mathcal{S}_{i,j}^k\}) + u_{i,j}^k \\ &= u_{i,j}^k + \tau \begin{cases} g_{i,j}^k \mathcal{D}_{i,j}^k + (1 - g_{i,j}^k)\mathcal{D}_{i,j}^k, & \mathcal{D}_{i,j}^k < \mathcal{S}_{i,j}^k \\ g_{i,j}^k \mathcal{S}_{i,j}^k + (1 - g_{i,j}^k)\mathcal{S}_{i,j}^k, & \text{else} \end{cases} \\ &= \begin{cases} u_{i,j}^k + \tau \mathcal{D}_{i,j}^k, & \mathcal{D}_{i,j}^k < \mathcal{S}_{i,j}^k \\ u_{i,j}^k + \tau \mathcal{S}_{i,j}^k, & \text{else} \end{cases} \\ &\geq \min_{n,m \in \Gamma} \{f_{n,m}\} \end{aligned} \tag{8.14}$$

8.3 Evolving Guidance Term

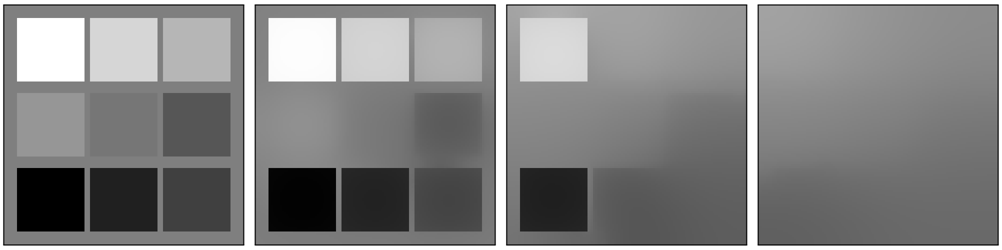
Using an evolving guidance image in the shock term leads to a shock-diffusion equation that looks similar to nonlinear isotropic diffusion, especially if the second derivative in gradient direction is used in the guidance term. Not only the equation, but also the behavior of the shock-diffusion filter is similar.
Figure 8.2 compares the effects of shock-diffusion for different derivative operators. Like in the previous chapters, the differences between the derivative operators are visible. Compared to the images of the pure shock filter with evolving guidance image, the results seem more clean, since the diffusion removed small scale details. Especially for the second derivative in gradient direction and the Laplacian, this leads to a more segmented result. The second derivative in the dominant eigendirection of the structure tensor clearly shows its directional dependency. Especially the derivative in gradient direction leads to a result that resembles the Perona Malik Diffusion. However, in the direct comparison to nonlinear diffusion the edges of structures in the shock-diffusion results are a lot sharper and the resulting images look more segmentation like.



(a) Initial image (b) Diffusion

(c) $Lu = \Delta u$ (d) $Lu = L^{\nabla} u$ (e) $Lu = L^{\nabla} u$

Figure 8.2: Shock-diffusion with an evolving guidance image compared to nonlinear isotropic diffusion. (a) Initial image. (b) Isotropic nonlinear diffusion with Peronal-Malik diffusivity, $\lambda = 1$. (c) Laplacian, $Lu = \Delta u$. $T = 500$ $\lambda = 1$. (d) Second flow line derivative, $Lu = L^{\nabla} u$. $T = 500$ $\lambda = 1$. (e) $Lu = L^{\nabla} u$. $\rho = 3$ $\lambda = 1$ $T = 500$. For all images $\tau = 0.25$ and $\zeta = 0.5$ was used.



(a) Initial image. (b) $T = 100$ (c) $T = 1000$ (d) $T = 2500$

Figure 8.3: Shock-diffusion evolution at different stopping times using the Perona-Malik diffusivity. Shock term: $L^{\nabla}u$. $\zeta = 0$, $\tau = 0.25$ and $\lambda = 3$.

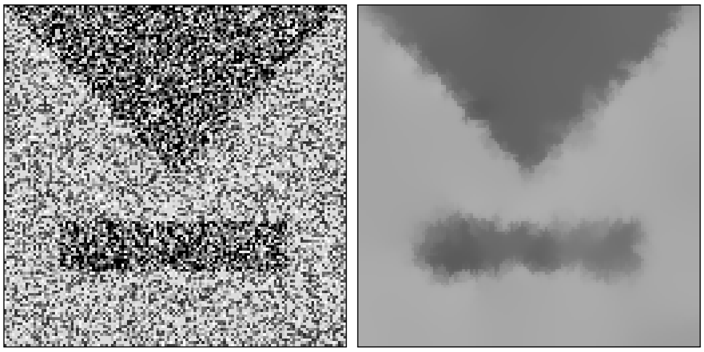
8.3.1 Steady State

Another similarity between shock-diffusion with evolving guidance image and Nonlinear Isotropic Diffusion is their steady state. For an increasing evolution time, the results of shock-diffusion becomes more blurry and even objects with a larger contrast vanish. Figure 8.3 visualizes this behavior. For a small stopping time, the objects with low contrast are removed. With a larger stopping time, higher contrast objects are removed. In the steady state, the resulting image would be a flat grey value. However, other than for diffusion, the grey value is not necessarily equal to the initial average grey value, since shock-diffusion does not have an average grey value invariance.

8.3.2 Denoising

Noise is a type of image degradation that may be created by the photo-sensors during image acquisition, but there are also other causes. In a noisy image, the grey values are changed, often without reference to the initial image. Since noise may change the grey values drastically, it can also influence the derivatives of the images heavily and with that also the result of shock filters. Moreover, removing the noise from the image (denoising) is a popular task for smoothing methods like diffusion. Therefore, we also test the performance of our shock-diffusion filters with respect to denoising.

Figure 8.4 demonstrated the denoising capabilities of shock-diffusion with an evolving, non-smoothed guidance image. The filter manages to remove the noise, especially in regions. Across edges, the smoothing effect is relatively small. The edges are not very clean, even after denoising. Since the diffusion propagates grey values the resulting regions are less bright/dark than they originally were.



(a) Noisy image. (b) $Lu = L^{\nabla}u$

Figure 8.4: Denoising with shock-diffusion with an evolving, non-smoothed guidance image. (a) Initial, noisy image. (b) Result of the shock-diffusion filter at $T = 62.5$ with guidance term $L^{\nabla}u$, $\zeta = 5$, $\tau = 0.25$ and $\lambda = 1$. The Charbonnier diffusivity is used as the weighting function.

8.4 Gaussian-smoothed Guidance Image

Similar to the classic shock filter, we can also add smoothing to the guidance image for our shock-diffusion operator. As we saw in the previous chapter, this smoothing leads to the creation of structures depending on which derivative operator is used. As the Laplacian does not seem to create useful structures, we are only using the directional derivatives in direction of the gradient and the dominant eigenvector of the structure tensor in the shock term. We will refer to the shock-diffusion that uses the shock term from the Coherence Enhancing Shock Filter as coherence enhancing shock-diffusion.

Shock filters that use a smoothed directional derivative in direction of the dominant eigenvector of the structure tensor create structures in a directionally dependent way. Using such a shock term in shock-diffusion leads to a mainly directionally dependent behavior of the filter near edges. This resembles the behavior of anisotropic diffusion methods like EED. Therefore, we compare the results of the shock-diffusion to the result created by EED in Figure 8.5. In the result of EED, small scale details with a small contrast, like the texture of the hat, are not visible anymore. Larger and higher contrast regions persist. Likewise, both results of the shock-diffusion filter remove such small scale details and smooth within the lager regions. The edges created by the shock-diffusion are much sharper than the edges created by EED and the result looks more segmentation-like. The implementation of EED was provided by Joachim Weickert.

Shock-diffusion has some advantages over EED other than sharper edges.

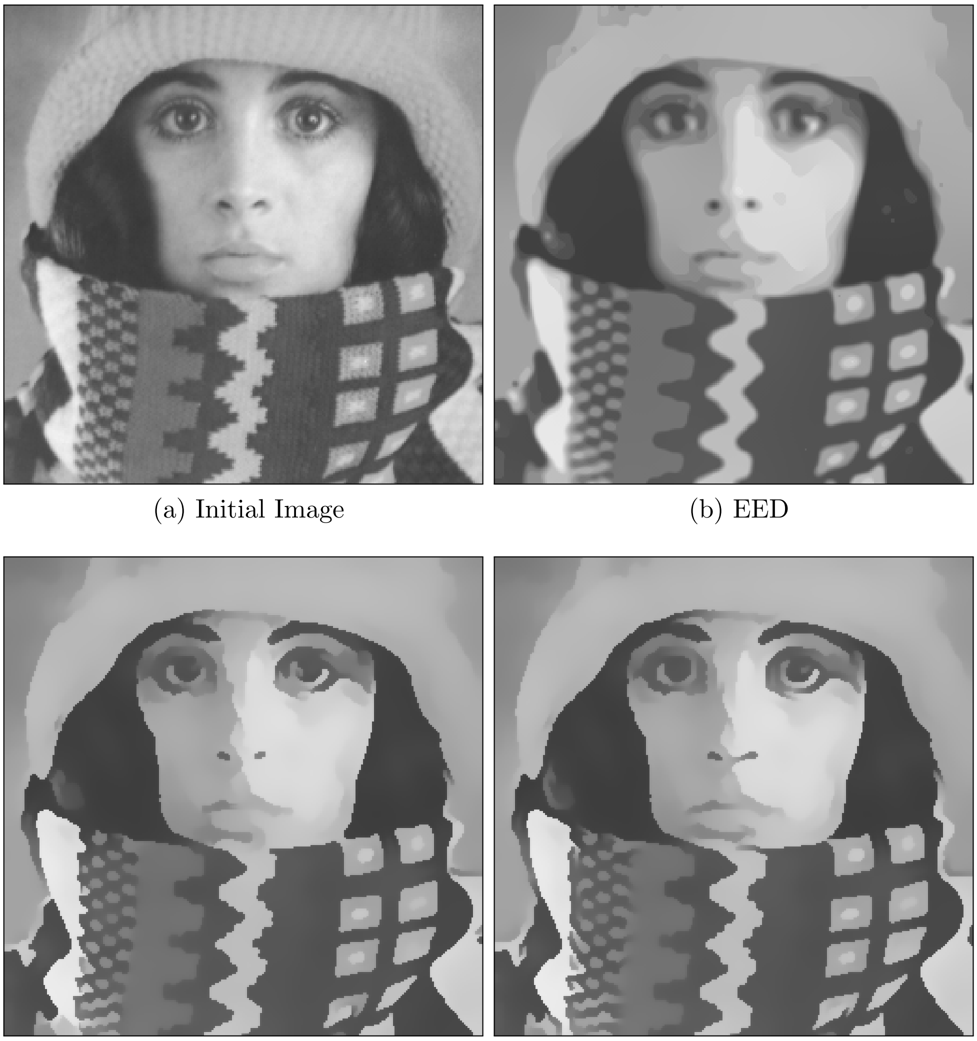
One problem of EED is that it is relatively hard to find a good discretization that does not violate a discrete minimum-maximum principle as stated by Weickert in a personal communication [36]. Numerical issues like this may lead to over- and undershoots and may cause edges to be less sharp. The shock-diffusion filter has a relatively simple numerical scheme that is easy to implement and even satisfies a discrete minimum-maximum principle. However, shock-diffusion also has a disadvantage. For EED, several theoretical properties such as well-posedness, average grey value invariance and many more have been shown [35]. For shock-diffusion, no such theoretical results are available, since they are already missing for shock filtering.

8.4.1 Denoising

Like before, we want to test the performance of the shock-diffusion with a Gaussian-smoothed guidance image in denoising. Interestingly, denoising can be used very well to explain the behavior of the filter. The difference between the two derivative operators is also highlighted.

In Figure 8.6, we denoised the same image as before using shock-diffusion with a Gaussian-smoothed guidance image. The middle row shows the result of using $L^{\nabla} u_{\sigma}$ as the derivative operator at different stopping times. The bottom row uses $L^{w}_{\rho} u_{\sigma}$ as the derivative operator. At the stopping time of $T = 2.5$, one can already see an effect. It smoothes primarily inside regions and not so much across edges. Looking at the results at $T = 100$ shows that the filter has smoothed the values inside the regions. Around the edges of the shapes a bright/dark regions forms. Its size corresponds to the size of the smoothing scale ζ that is used in the diffusivity. The values inside the regions of the shapes are still brighter than the pixels close to the edge and the values in the background are still darker than the values close to the edges. It seems like the shock term is trying to preserve the values close to the edge, while removing unwanted noise and sharpening the edge. At $T = 25000$, the values inside the regions are closer to the values at the edges. The diffusion propagates the values from the edges to the flat regions creating darker shapes and a brighter background. At $T = 37500$, the grey values inside the regions fit the grey values at the edges. Since there is a small amount of diffusion across edges, the values near the edges become slightly darker/brighter than they originally were, which leads to a slight darker background and slightly brighter shapes overall. Applying the shock-diffusion longer will eventually lead to a flat image.

The behavior displayed over time resembles inpaiting. Inpainting is an image processing task in which a filter needs to fill gaps using information from known pixels. We cover this topic in more detail in Chapter 9.1. Since the



(a) Initial Image (b) EED

(c) Shock-diff., $Lu = L^{\nabla} u$ (d) Shock-diff., $Lu = L^{w}_{\rho} u$

Figure 8.5: Comparing EED to shock-diffusion using the smoothed gradient derivative and the shock term from the Coherence Enhancing Shock Filter. The shock-diffusion uses the Charbonnier Diffusion. (a) Initial image. (b) EED uses a diffusivity that decreases exponentially in $|\boldsymbol{\nabla} u_{\zeta}|^2$ by Weickert [31], $\zeta = 0.5$ $\lambda = 2$ $T = 100$. (c) Flow Line Derivative, $\lambda = 1.5$ $\sigma = 1$ $\zeta = 1$, $T = 20$. (d) Coherence Enhancing Shock Filter, $\lambda = 1.5$ $\sigma = 1$ $\zeta = 1$ $\rho = 1.5$, $T = 20$.

values at the edges change very slowly, the diffusion almost inpaints the flat regions in the color close to the edges. Since the values close to the edges are not fixed, there is a difference to classical diffusion-based inpainting, but there is a relation.
Comparing the result of the derivative in direction of the gradient and the eigenvector of the structure tensor shows, that the estimation by the eigenvector of the structure tensor seems closer to the original. The operator $Lu = L^w_\rho u_\sigma$ seems to be more robust against noise than $L^\nabla u_\sigma$

8.5 Creating a Non-flat Steady State

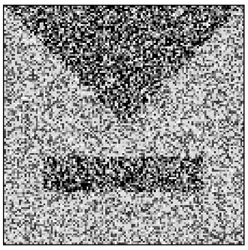
Diffusion and shock filtering are opposing processes: diffusion blurs edges, while shock filtering sharpens them. Nevertheless, the steady state of shock-diffusion always seems to be flat. We first explain, why shock-diffusion typically results in a flat steady state.
Consider the following shock-diffusion evolution:

$$u_t = g\Delta u - (1-g)\mathrm{sgn}(\Delta u)|\boldsymbol{\nabla} u| \tag{8.15}$$

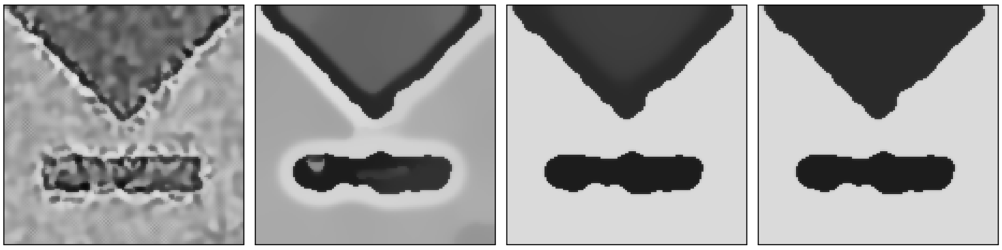
So both processes always have opposite signs. If $\Delta u > 0$ then the sign of the shock filter is negative and vice versa for $\Delta u < 0$. So the absolute values of the shock filter and the diffusion term determine whether a value becomes larger or smaller. Imagine a convex region. Here the shock term will try to shrink values. Since the region is convex, $\Delta u > 0$. So the diffusion tries to enlarge values in that region. So pixels with a larger Laplacian than gradient magnitude become larger instead of smaller.
Unfortunately, multiplying the shock term with a scale to enlarge its influence does not solve the problem. Consider the one dimensional signal f and its derivatives:

$$\begin{aligned} f &= (0,1) \\ f_{xx} &= (1,-1) \\ |f_x| &= (0,0) \end{aligned}$$

where $|f_x|$ is computed using the Roy-Tourin upwind sheme. As long as a weighting function is used that can not become 0, the diffusion part of the shock-diffusion will always blur this edge until the grey values are equal, since the pixel remains a local minimum/maximum even after diffusion.
So the only guaranteed possibility to create a non-flat steady state, is choosing a weighting function that can become 0.



(a) Noisy Image

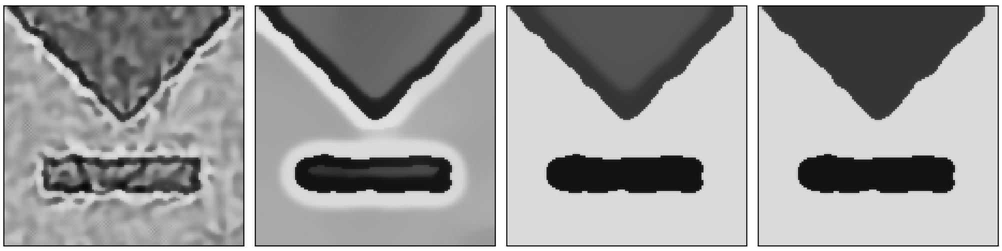


(b) $Lu = L^{\nabla} u_{\sigma}$, $T = 2.5$

(c) $Lu = L^{\nabla} u_{\sigma}$, $T = 100$

(d) $Lu = L^{\nabla} u_{\sigma}$, $T = 25000$

(e) $Lu = L^{\nabla} u_{\sigma}$, $T = 37500$



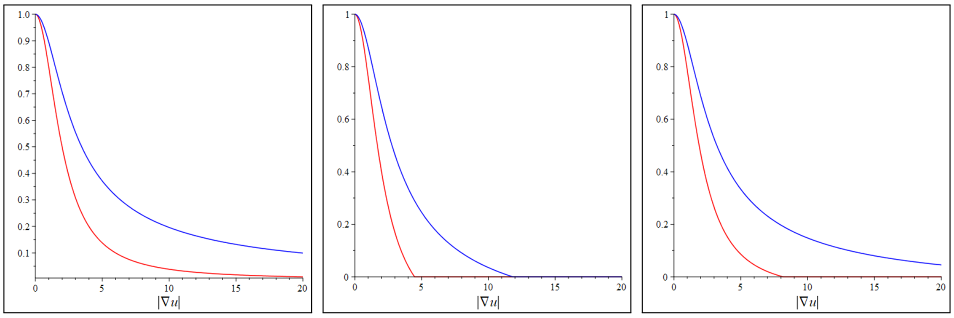
(f) $Lu = L^{w}_{\rho} u_{\sigma}$, $T = 2.5$

(g) $Lu = L^{w}_{\rho} u_{\sigma}$, $T = 100$

(h) $Lu = L^{w}_{\rho} u_{\sigma}$, $T = 25000$

(i) $Lu = L^{w}_{\rho} u_{\sigma}$, $T = 37500$

Figure 8.6: Denoising with shock-diffusion with a Gaussian-smoothed guidance image. (a) Noisy image. (b)-(i) shock-diffusion evolution at different stopping times. Middle row: $Lu = L^{\nabla} u_{\sigma}$, $\sigma = 1.5$. Bottom row: $Lu = L^{w}_{\rho} u_{\sigma}$ with $\sigma = 1.6$ and $\rho = 2.05$. Both evolutions use the Charbonnier diffusivity and $\tau = 0.25$, $\lambda = 1$ and $\zeta = 5$.



(a) Perona-Malik and Charbonnier diffusivity (b) $\alpha = 0.2$ (c) $\alpha = 0.06$

Figure 8.7: Effect of α in modified diffusivities. (a) Perona Malik (red) and Charbonnier (blue) diffusivity. (b)-(c) Modified Perona Malik (red) and Charbonnier (blue) diffusivity with $\lambda = 2$.

8.5.1 Modified Diffusivity

Diffusivities never become 0, since this is needed for theoretical properties of diffusion [35]. Therefore, we propose modifying diffusivities. For the Perona-Malik diffusivity, the modification looks as follows:

$$g_{PM}^{m}(s^2) = \max\left\{\frac{1+\alpha}{1+\frac{s^2}{\lambda}} - \alpha, 0\right\} \tag{8.16}$$

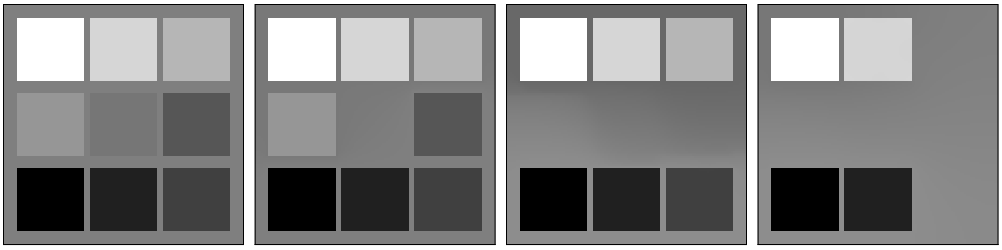
The Charbonnier diffusivity is modified similarly:

$$g_{CH}^{m}(s^2) = \max\left\{\frac{1+\alpha}{\sqrt{1+\frac{s^2}{\lambda}}} - \alpha, 0\right\} \tag{8.17}$$

Figure 8.7 shows the graph of the modified diffusivities for different values of α. The parameter α determines if the diffusivity is set to 0.

In Figure 8.8, the steady states of the shock filters with modified Perona-Malik diffusivity of the palette image are shown for different values of α. The parameter determines which squares still exist in the steady state. The squares with a higher contrast survive even for small values of α, while squares with a lower contrast vanish for a smaller α.

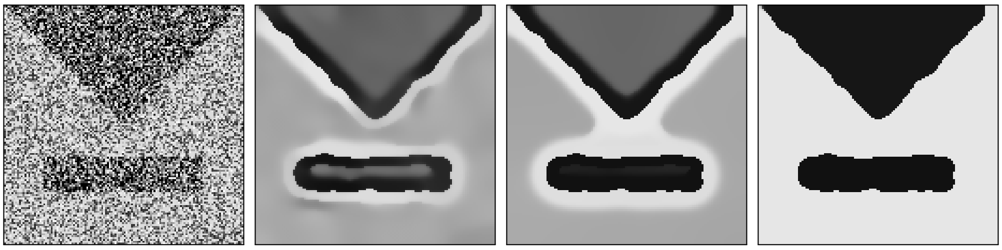
In the previous section, we explained that the shock term is aims to fix the values close to edges, while the diffusion term also smoothes across edges if a diffusivity is used. This conflict does not arise if a modified diffusivity is used. Values that are close to an edge with a sufficiently large gradient



(a) Initial Image (b) $\alpha = 0.02$ (c) $\alpha = 0.006$ (d) $\alpha = 0.003$

Figure 8.8: Steady states of shock-diffusion for different α. The shock term uses the second derivative in gradient direction. $\tau = 0.25$, $\lambda = 3$ and $\sigma = 0$ for all resulting images. The modified Perona-Malik diffusivity is used.

magnitude are changed only by the shock term. However, the edge does not have to exist in the beginning of the evolution, since the shock filter may elongate or create edges. Once such an edge with a sufficiently large gradient magnitude has formed, the shock filter preserves that edge. The diffusion can not influence the edge. Instead, it propagates values from the edge into the region adjacent to such an edge. That means the values near edges act as an inpainting mask (known values in inpainting) and the diffusion then inpaints regions surrounded by edges using the values at the fixed edges.
This behavior can be visualized very well using the denoising example from the previous section. In Figure 8.9, a similar experiment is performed. We denoise the same image using shock-diffusion with a modified diffusivity. In early iterations, the diffusion term smoothes inside regions and slightly across edges and the shock term enhances the edges. Since the shock term from the Coherence Enhancing Shock Filter is used, it may even fill gaps in edges. More importantly, the shock term keeps the values relatively close to the original values. At $T = 250$, you can already see the inpainting take effect. Since the edges can not be changed by the diffusion anymore, the grey values at edges are fixed and act as a mask for inpainting by the diffusion term. The rectangle is already filled with relatively dark values and the background starts to become bright. In the steady state, the dark shapes and the bright background are clearly distinguishable.



(a) Initial Image (b) $T = 25$ (c) $T = 250$ (d) Steady state

Figure 8.9: Denoising with shock-diffusion with modified Charbonnier diffusivity. (a) Initial image. (b)-(d) Results for different stopping times. $Lu = L^w_\rho u_\sigma$, $\sigma = 1.6$, $\rho = 2.05$, $\zeta = 5.5$, $\lambda = 1.5$ and $\alpha = 0.15$.

Chapter 9

Inpainting with Shock-Diffusion Filtering

We already mentioned the task of inpainting in the previous chapter. There, we explained that the behavior of shock-diffusion resembles inpainting, depending on the chosen weighting function. In this chapter, we demonstrate that shock-diffusion can be used to perform image inpainting and even performs well, compared to well-proven methods like EED.

9.1 PDE-based Inpainting

In inpainting, the task is to fill the gaps within an image using the information given by the present pixels. We call the set of known pixels the interpolation mask or inpainting mask M. The set of unknown pixels is called the inpainiting domain $\Gamma \backslash M$. This task can be fulfilled using PDE-based methods.

A brief overview on this topic is provided by Schmaltz et al. in [27]. We use the model for inpainting from this publication. Formally, PDE-based inpainting can be described using the following PDE:

$$0 = (1 - c_M)Du - c_M(u - f) \tag{9.1}$$

where f is the initial image and Du the derivative operator that describes the inpainting operator. c_M is the characteristic function of M:

$$c_M(\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{x} \in M \\ 0, & \text{else} \end{cases} \tag{9.2}$$

Since the inpainting does not change the values at the locations inside the interpolation mask, the equation can be simplified. On $\Gamma \backslash M$, the solution of

Equation (9.1) is given by the steady state of

$$u_t = Du \tag{9.3}$$

The image in the steady state of that evolution corresponds to the final result of the inpainting process.

There is a large number of possible choices for the inpainting operator Du. Schmaltz et al. provide a comparison of the performance of serveral different inpainting operators in [27]. The best results in their different experiments are achieved by using EED as the inpainting operator. Therefore, we compare the performance of shock-diffusion and EED in inpainting in the following sections. For the implementation of EED, we use an implementation that Joachim Weickert provided to us and modified it to use the Charbonnier diffusivity in the construction of the diffusion tensor.

9.2 Shock-Diffusion Inpainting Operator

Shock-diffusion, as we defined it in the previous section, allows different shock terms and weighting functions. For the inpainting operator, we use the two shock terms that gave the best results in the previous section. The first operator uses the second derivative in direction of the gradient of the smoothed, evolving guidance image:

$$Du = g(|\boldsymbol{\nabla} u_\zeta|^2)\Delta u - (1 - g(|\boldsymbol{\nabla} u_\zeta|^2))\mathrm{sgn}(L^{\nabla} u_\sigma)|\boldsymbol{\nabla} u| \tag{9.4}$$

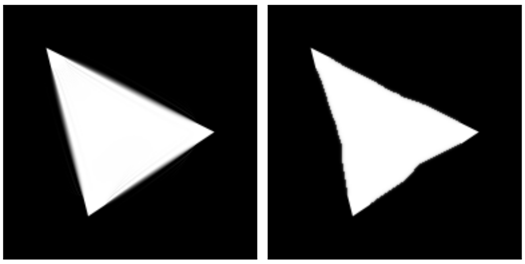
The second operator uses the shock term from the Coherence Enhancing Shock Filter:

$$Du = g(|\boldsymbol{\nabla} u_\zeta|^2)\Delta u - (1 - g(|\boldsymbol{\nabla} u_\zeta|^2))\mathrm{sgn}(L^{w}_{\rho} u_\sigma)|\boldsymbol{\nabla} u| \tag{9.5}$$

For all of the produced inpaiting results, we use the Charbonnier diffusivity as the weighting function.

To compare the performance of the two candidates for shock-diffusion based inpainting, we performed one of the experiments, that was performed by Schmaltz et al. to compare the performance of different inpainting operators in [27]. Figure 9.1 shows the results of that experiment. The initial image consists of three disks which contain white wedges on a dark background. Connecting the corners with straight lines yields a white triangle on a dark background. The data inside the disks is used as the interpolation mask. Everything outside the disks is supposed to be inpainted. The shock-diffusion using the coherence enhancing shock term reconstructs the triangle relatively well. The reconstructed edges are not as straight as the

(a) Initial Image (b) Nonlin. iso. diff. (c) Shock-diff. $L^{\nabla}u_{\sigma}$



(d) EED (e) Shock-diff. $L^{w}_{\rho}u_{\sigma}$

Figure 9.1: Comparison of different inpainting operators. (a) Initial image. (b) Nonlinear isotropic diffusion with Charbonnier diffusivity, $\lambda = 1$. (c) Shock-diffusion using the second derivative in gradient direction, $\lambda = 3$, $\zeta = 5.5$ and $\sigma = 1.75$. (d) EED, $\lambda = 0.01$ and $\zeta = 4$. (e) Shock-diffusion using the coherence enhancing shock term, $\lambda = 3$, $\rho = 5.5$, $\zeta = 5.5$ and $\sigma = 1.75$. We received the initial image and the result of EED from Joachim Weickert.

edges produced by EED. We suppose that this is related to the discretization of the shock term. The produced edges are very sharp.
The result that is produced by using the second derivative in gradient direction does not connect the corners. It strongly resembles resulting image produced by nonlinear isotropic diffusion, in which the corners are not connected either. This behavior is surprising, considering that shock-diffusion using both shock terms has similar results when they are used to segment or denoise images. However, looking at the results of the shock filter that uses $L^{\nabla}u_{\sigma}$ as its guidance term shows (see Ch. 7) that its behavior does not seem anisotropic. Weickert's Coherence Enhancing Shock Filter does behave anisotropic, even though the two shock filters only differ from each other by the choice of the direction for the directional derivative in the guidance term. We suppose that the difference in inpainting has the same cause. The directional dependency when using $L^{w}_{\rho}u_{\sigma}$ in shock-diffusion is caused by the choice of the directional derivative.

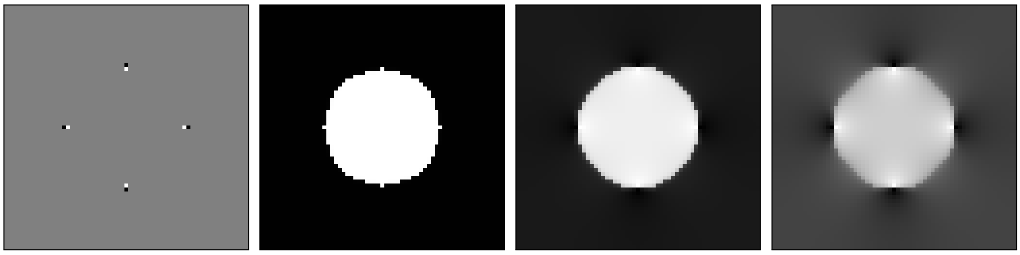
Since the coherence enhancing shock term performs better, we only consider the inpainting operator from Equation 9.5. This is not the first inpainting operator that uses morphological methods in combination with eigenvalues of the structure tensor. Like our operator, the method of Bornemann and März [4] uses derivatives in an eigendirection of the structure tensor and morphological methods for inpainting.

9.2.1 Parameters

The proposed inpainting operator has many different parameters. This has the advantage, that the process is relatively flexible. On the other hand, it is not always easy to choose the right parameters to achieve a good result.

Contrast Parameter

The shock-diffusion inpainting operator is not only a combination of shock filtering and diffusion, but also a combination of two operators that propagate information in their own way. Homogeneous diffusion propagates information equally strong in all direction and therefore it creates relatively blurry results. If used as an inpainting operator, the homogeneous diffusion creates grey values gradients, but it does not create sharp edges [5]. On the other hand, the Coherence Enhancing Shock Filter propagates information in the a direction that is determined by the structure tensor. While the filter thickens structures it does not freely propagate values in the other direction. The balance of these two processes is determined by the parameters that influence the weighting function, namely the presmoothing scale ζ and the contrast parameter λ. While the smoothing scale only has an indirect influence, the contrast parameter directly influences the weight of the respective operator. A very small choice of λ leads to a generally larger weight of the shock term. Therefore, the shock term propagates values according to its direction estimation until it reaches a state in which it can not change anything anymore. The only thing that happens after that, is the inpainting by the homogeneous diffusion. From the structures that were mainly created by the shock term, it inpaints the remaining regions.
A relatively large contrast parameter leads to an overall larger influence of the diffusion term. In this case, the operator almost acts like a restricted homogeneous diffusion inpainting operator. The diffusion is very strong and the shock term restricts it to some degree at edges. Therefore, the result is blurry, but less blurry than homogeneous diffusion without the restriction of the shock term.
Another option is choosing the value for λ neither very small nor very large.



(a) Initial Image (b) $\lambda = 0.5$ (c) $\lambda = 7$ (d) $\lambda = 10$

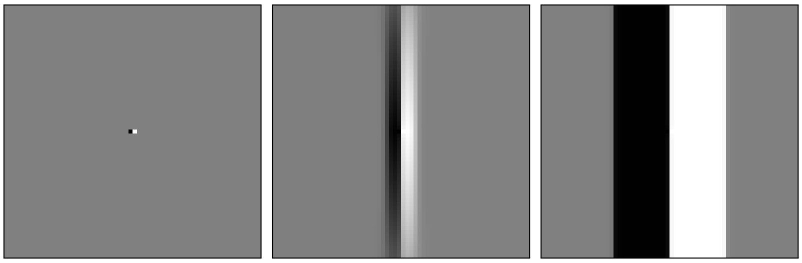
Figure 9.2: Inpainting with shock-diffusion for different contrast parameters. (a) Initial image. (b) $\lambda = 0.5$, $\sigma = 2.05$, $\rho = 3$ and $\zeta = 1$. (c) $\lambda = 7$, $\sigma = 1.9$, $\rho = 3$ and $\zeta = 1$. (d) $\lambda = 10$, $\sigma = 2.05$, $\rho = 3$ and $\zeta = 1$. The initial image was provided by Weickert.

This leads to the edges created by the shock term and the inpainting of the regions by the diffusion at a similar speed.

Figure 9.2 compares the result of shock-diffusion inpainting with the Charbonnier diffusivity for different values of λ. The initial image consists of four dipoles. The grey region is supposed to be inpainted. A small λ yields very sharp edges and a clear separation between the two regions. For a larger contrast parameter, the edges are not as sharp and the resulting regions are also not as clean. A really large λ leads to relatively blurry edges.
While all three evolutions reconstruct a similar shape, there are differences in the result. Clearly, the smallest λ yields the best result for this example. However, choosing the contrast parameter too small, may lead to unwanted effects. As one can see from the example in Figure 7.13 in Chapter 7, the Coherence Enhancing Shock Filter might not estimate the direction for the propagation correctly, if the given information is not adequate. The same holds true for the coherence enhancing shock term in the inpainting operator. So for inpainting problems, in which a good direction estimation is not possible from the start, it might create structures that do not fit the desired result. Therefore, choosing a slightly larger λ can yield better inpainting results, since the direction estimation can be improved. However, the larger λ comes with the cost of less sharp edges.

Smoothing Scale of the Shock Term

The smoothing scale σ of the shock term determines the scale of the structures that are constructed by the shock term. In a well inpainted result, it can not be detected after the inpainting. So to visualize the effect of different values of σ, we do not look at the steady states of the inpainting process in



(a) Initial Image (b) $\sigma = 1$ (c) $\sigma = 5$

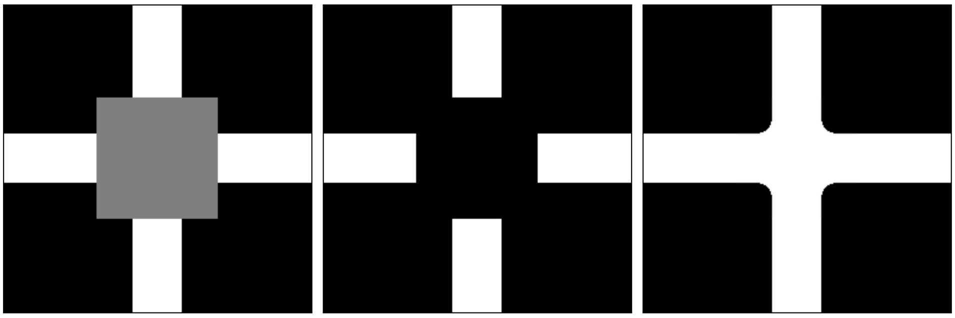
Figure 9.3: Effect of different σ on shock-diffusion inpainting. (a) Initial image. (b) - (c) $\lambda = 1$, $\rho = 3$, $\zeta = 3$ and $T = 1250$. The initial image was provided by Weickert.

Figure 9.3. The initial image is one dipole of a black and white pixel. From this dipole the inpainting process creates an edge in the middle of the image. On the left of the edge, there is a dark bar and a bright bar on the right. The thickness of this bar is determined by the parameter σ. For a large σ the bar is thicker. This is the behavior that can also be observed from the Coherence Enhancing Shock Filter, so it is not surprising that the parameter has this effect.

Ahead-Looking Parameters

The integration scale ρ and the smoothing scale ζ of the diffusivity are both used to average local gradient information. In a way, they look ahead to determine the structure outside the inpainting mask and the already inpainted pixels. The smoothing scale influences the scale on which the weight of shock and diffusion term is computed. The integration scale determines the local structure to estimate the direction of the created edges.
Both parameters are similar, but they are used in different contexts. Since ζ determines whether the weight of the shock filter should be large, choosing similar values for integration scale and smoothing scale typically works well.

Since the two parameters look ahead of the already inpainted data, it has a large influence on the result of the inpainting. The choice of these parameters can determine whether objects are connected by the inpainting and how they are connected. In Figure 9.4, we can see the effect of smaller and larger values of the two parameters. The initial image depicts a cross that is missing the crossing of the horizontal and vertical line. For smaller values of ρ and ζ, the bars are not connected to a cross. For sufficiently large values, the inpainting creates the crossing.



(a) Initial Image (b) $\rho = 1.5$, $\zeta = 1.5$ (c) $\rho = 3$, $\zeta = 2$

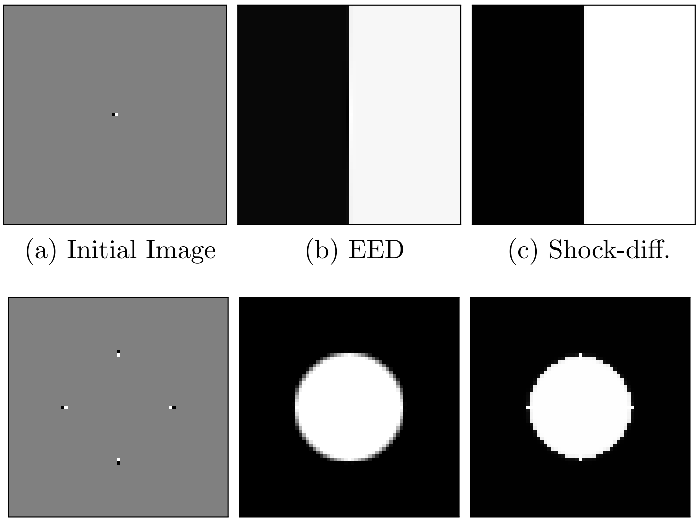
Figure 9.4: Effect of different ζ and ρ on shock-diffusion inpainting. (a) Initial image. (b) - (c) Result of shock-diffusion $\lambda = 2$ and $\sigma = 2$. The initial image was provided by Weickert.

9.3 Comparison to EED

EED is a very successful inpainting operator that produces high quality results. Therefore, it has even been implemented into the image compression framework R-EED, which has been shown to outperform other popular image compression codecs such as JPEG and JPEG2000 [27]. One strength of EED is that it can be used for the reconstruction of shapes from very sparse data sets. One example of that was already given in Figure 9.1. In one of his talks [34], Weickert presented a number different experiments that show how well EED performs in reconstructing shapes from a very sparse data set. Using these examples, we show that shock-diffusion with a coherence enhancing shock term can achieve results of a similar quality. To compare our results to the result created by EED, Joachim Weickert provided us the images that show the inpainting result by EED, as well as the input images.

Figure 9.5 demonstrates the ability of both EED and shock-diffusion to reconstruct a shape from only few dipoles. One of the initial images uses only one dipole as known data. Both inpainting operators reconstruct a segmentation of the image into a black and white plane. The created edge between the planes in both resulting images is really sharp and clean. The other initial image has four dipoles. EED and shock-diffusion reconstruct a white disk on a black background from these four dipoles. The outline of the disk looks slightly different. The result created by EED has a very smooth outline. Like the Coherence Enhancing Shock Filter, the shock term of shock-diffusion creates structures of one value, which is a minimal or maximal value. Therefore, the outline of the disk is not smooth, but slightly pixelated.

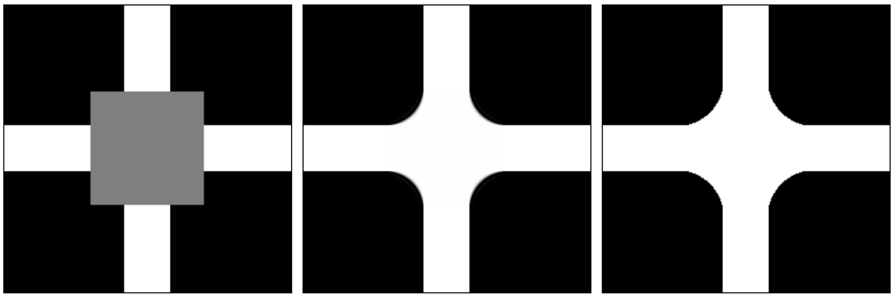
Figure 9.6 compares the capability of shock-diffusion and EED to restore



(a) Initial Image (b) EED (c) Shock-diff.

(d) Initial Image (e) EED (f) Shock-diff.

Figure 9.5: Inpainting from one dipoles using EED and shock-diffusion. (a) Initial image: one dipole. (b) EED, $\zeta = 1$ and $\lambda = 0.01$. (c) Shock-diffusion, $\lambda = 1$, $\rho = 2.3$, $\zeta = 1$ and $\sigma = 1$. (d) Initial image of four dipoles: (e) EED, $\zeta = 1$ and $\lambda = 0.01$. (f) Shock-diffusion, $\lambda = 2$, $\rho = 2$, $\zeta = 2$ and $\sigma = 2.05$. The images created by EED were provided by Joachim Weickert.



(a) Initial Image (b) EED (c) Shock-diff.

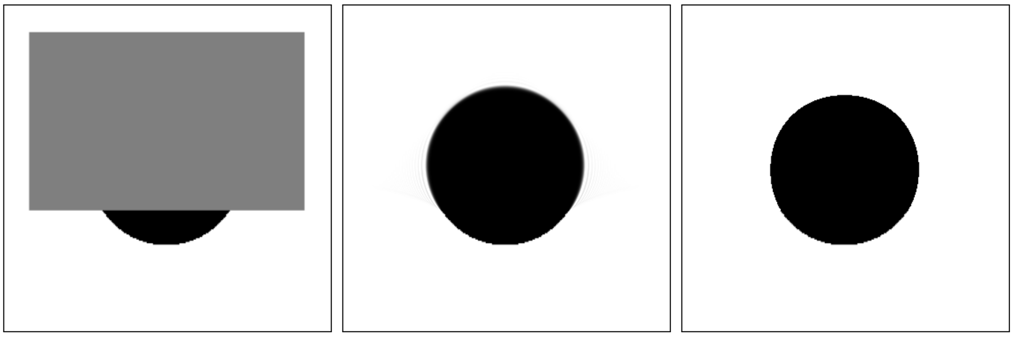
Figure 9.6: Restoration of a cross with EED and shock-diffusion. (a) Initial image. (b) EED, $\zeta = 3$ and $\lambda = 0.003$. (c) Shock-diffusion, $\lambda = 2$, $\rho = 3$, $\zeta = 3$ and $\sigma = 3$. The initial image and the image created by EED was provided by Joachim Weickert.

a cross from an image of a cross that is missing its crossing in the center. We already used this initial image in the previous section. Here, the parameters of the shock-diffusion are chosen, such that the result matches the result produced by EED. The produced results of both operators look really similar. In both reconstructions, the four bars are connected, such that a cross with rounded corners is formed. A close look at the rounded corners shows that EED produces a smooth edge, while shock-diffusion creates the slightly pixelated effect.

Figure 9.7 shows the reconstruction of a disk by EED and shock-diffusion from a heavily occluded disk. The initial image shows the bottom part of a disk. The grey rectangle is supposed to be inpainted to reconstruct the missing part of the disk. Both operators manage to reconstruct the disk. In this example, one can see that the shock-diffusion inpainting creates slightly sharper edges than EED.

In Figure 9.8, compares the result of inpainting with EED and shock-diffusion for a challenging inpainting task. The original image depicts a cat. Points that are important for the shape of the cat, are preserved in the disks in the inpainting mask. Everything outside the disks is supposed to be inpainted. EED almost perfectly reconstructs the cat. The inpainting result of shock-diffusion is close to the result of EED. For example, the head or the back of the cat of both results look similar. The reconstruction of the foot by shock-diffusion is slightly curved instead of straight as in the original image.

The previous examples show that the performances of EED and shock-diffusion in reconstruction shapes from very sparse data sets are comparable.



(a) Initial Image (b) EED (c) Shock-diff.

Figure 9.7: Reconstruction of an occluded disk with EED and shock-diffusion. (a) Initial image. (b) EED, $\zeta = 20$ and $\lambda = 0.0003$. (c) Shock-diffusion, $\lambda = 1$, $\rho = 3$, $\zeta = 2$ and $\sigma = 70$. The initial image and the image created by EED was provided by Joachim Weickert.

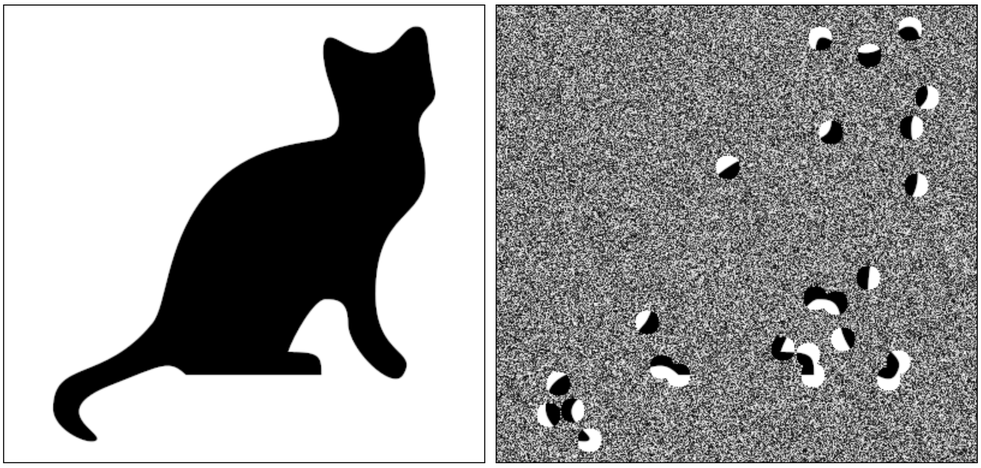
Due to the properties of the shock term, shock-diffusion can produce sharper edges than EED. Shock-diffusion even creates comparable result for difficult inpainting tasks like the reconstruction of the cat image. Unfortunately, it also produces some errors like the bend foot of the cat in Figure 9.8 or the imperfect edges of the triangle from Figure 9.1.

9.4 Natural Images

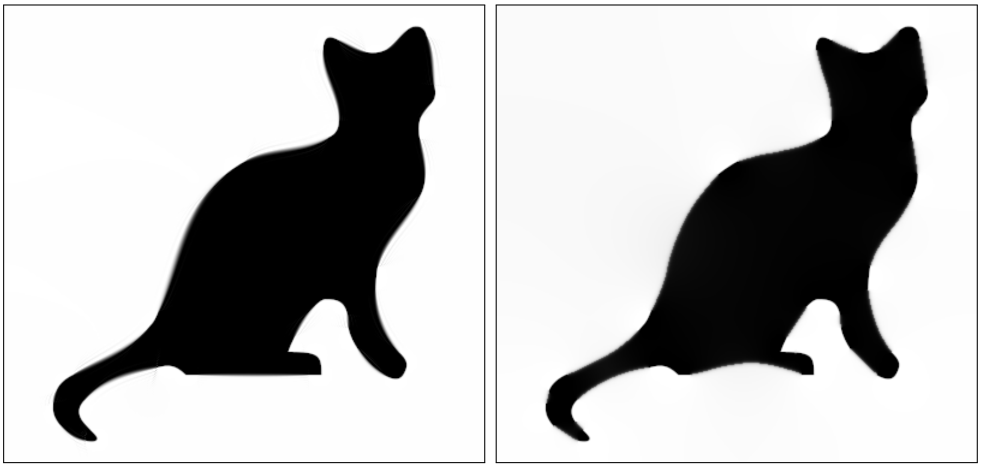
Inpainting is not only used to reconstruct shapes from sparse data, but often also to reconstruct natural images. For example, in the image compression codec R-EED natural images are reconstructed using EED as the inpainting operator [27]. Therefore, we investigate the result of shock-diffusion inpainting using some experiments. In our experiments, we use inpainting masks with a random distribution of pixels. Code to produce such masks was provided to us by Joachim Weickert.
To determine the quality of the inpainting, we compare the result of the inpainting process to the original image using the mean squared error (MSE). The mean squared error penalizes differences between the value of two pixels in a quadratic manner. It is defined as follows:

$$MSE(f,g) = \frac{1}{n_x n_y} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} (f_{i,j} - g_{i,j})^2 \tag{9.6}$$

Figure 9.9 compares the inpainting result of EED and shock-diffusion for



(a) Original Image (b) Inpainting mask



(c) EED (d) Shock-diff.

Figure 9.8: Reconstruction of a cat from a sparse inpainting mask. (a) Original image. (b) Inpainting mask. (c) EED, $\zeta = 4$ and $\lambda = 0.01$. (d) Shock-diffusion, $\lambda = 7.2$, $\rho = 6$, $\zeta = 4.5$ and $\sigma = 4$. The initial image and the image created by EED was provided by Joachim Weickert.

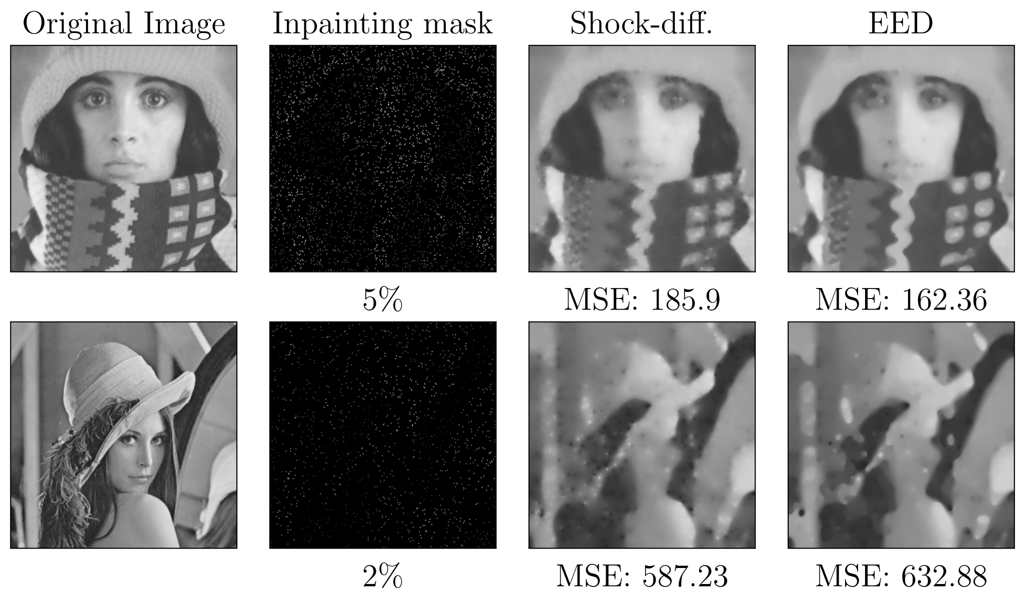


Figure 9.9: Reconstruction of a natural image from randomly chosen pixels using shock-diffusion and EED. Top row: 5% of the original pixels. Shock-diffusion with MSE of 185.9 $\sigma = 1.45$, $\rho = 4.9$, $\zeta = 4.9$ and $\lambda = 6.1$. EED with MSE of 162.36 $\lambda = 0.5$ and $\zeta = 1$. Bottom row: 2% of the pixels. Shock-diffusion with MSE of 587.23 $\sigma = 2$, $\rho = 6$, $\zeta = 5$ and $\lambda = 6$. EED with MSE of 162.36 $\lambda = 0.5$ and $\zeta = 1$.

spare data of natural images. At 5% of the original pixels, the results of both operators are have a similar visual quality, even though the result of EED has a lower MSE compared to the original image. For even less pixels, both operators still perform similarly. The inpainting by the shock-diffusion operator even has a smaller MSE than the result produced by EED. So even when inpainting a natural image from very sparse and randomly selected data, shock-diffusion can keep-up with the performance of EED.

Chapter 10

Conclusion and Outlook

10.1 Conclusion

In this thesis, we have investigated several aspects of shock filtering. We gained a better understanding of shock filtering by decoupling the guidance image from the evolving image. We proved, that in case of a fixed guidance image, all shock filters behave equally and result in the same steady state in strictly convex and strictly concave regions. Although we can not make such a strong statement in case of an evolving guidance image, we explained that the behavior of shock filters with different derivative operators is still similar to the fixed case. We used these findings to explain the finding of Kramer and Bruckner [16], that shock filters perform well for deblurring-cartoon like images.

Shock filters often suffer from over-segmentation and are noise sensitive. This problem has been addressed by presmoothing the evolving guidance image with Gaussian convolution in [2]. Using different examples, we showed that this presmoothing does not necessarily lead to a regularized shock filter. It has the interesting effect of creating or removing structures of the same scale as the Gaussian, that is used for the presmoothing. We explained that this effect is linked to the equivalence of homogeneous diffusion and Gaussian convolution, and the property of diffusion to propagate information throughout an image.

Furthermore, we proposed shock-diffusion as an extension to shock filtering. A shock-diffusion filter combines homogeneous diffusion and shock filtering by using a weighted sum of the corresponding shock and diffusion term. Unlike other combinations of smoothing and shock filtering, our operator is stable with respect to a discrete minimum-maximum principle. Our experiments suggested, that shock-diffusion with diffisivites as weighting functions behave similar to diffusion. Shock-diffusion produces the best result using the coherence enhancing shock term. Using the proposed modified diffusivites as the weighting functions may lead to a non-flat steady state.

We made use of the similarity to diffusion filters, by applying the shock-diffusion operator that uses the coherence enhancing shock term to inpainting problems. The comparison of our operator to EED suggests, that shock-diffusion can produce results with a similar quality to EED, in shape reconstruction and inpainting of natural images from sparse data.

10.2 Future Work

Our shock-diffusion operator produced qualitatively good results. It creates sharp edges and is highly flexible. Unfortunately, the estimation of the direction of edges is not always optimal. Therefore, we will try to improve the discretization of the shock term.
EED has successfully been used for image compression in the R-EED codec [27]. Since our inpainting operator can produce a quality that is similar to EED, we want to evaluate its performance in an image compression framework. The distance between the data points in the interpolation mask seems to be relevant for the result of shock-diffusion. Thus, different data selection strategies may be more successful than the one used in the R-EED codec for shock-diffusion.

Chapter 11

Appendix

Shock Filter with evolving guidance image in the strictly concave regions

If $(u_{pp})^k_{i,j} < 0$, then image at the time $k+1$ is then given by

$$u^{k+1}_{i,j} = u^k_{i,j} + \tau\Bigg(\max\Big\{0, \frac{u^k_{i+1,j} - u^k_{i,j}}{h_x}, \frac{u^k_{i-1,j} - u^k_{i,j}}{h_x}\Big\}^2 + \max\Big\{0, \frac{u^k_{i,j+1} - u^k_{i,j}}{h_y}, \frac{u^k_{i,j-1} u^k_{i,j} - u^k}{h_y}\Big\}^2\Bigg)^{\frac{1}{2}} \tag{11.1}$$

Without loss of generality, we assume that in both x- and y-direction the backward forward difference is larger and that $|u^k_{i,j} - u^k_{i+1,j}| \geq |u^k_{i,j} - u^k_{i-1,j}|$. Therefore, the following holds:

$$\begin{aligned} u^{k+1}_{i,j} &= u^k_{i,j} + \tau\sqrt{\Big(\frac{u^k_{i+1,j} - u^k_{i,j}}{h_x}\Big)^2 + \Big(\frac{u^k_{i,j+1} - u^k_{i,j}}{h_y}\Big)^2} \\ &\leq u^k_{i,j} + 2\tau\frac{u^k_{i+1,j} - u^k_{i,j}}{h_x} \\ &= u^k_{i,j}\Big(1 - 2\frac{\tau}{h_x}\Big) + \frac{2\tau}{h_x} u^k_{i+1,j} \end{aligned} \tag{11.2}$$

$$\tag{11.3}$$

For the second directional derivative in the normalized and arbitrary direction $\boldsymbol{p}$ at time level $k+1$, the following holds:

$$(u_{pp})^{k+1}_{i,j} = \Big(p_1^2 \frac{u^{k+1}_{i+1j} - 2u^{k+1}_{i,j} + u^{k+1}_{i-1j}}{h_x^2} + p_2^2 \frac{u^{k+1}_{ij+1} - 2u^{k+1}_{i,j} + u^{k+1}_{ij-1}}{h_y^2}$$

$$
\begin{aligned}
&+ 2p_1p_2\frac{u^{k+1}_{i+1j+1} + u^{k+1}_{i-1,j-1} - u^{k+1}_{i+1j-1} - u^{k+1}_{i-1j+1}}{4h_xh_y}\Big)\\
&\leq (1-2\frac{\tau}{h_x})\Big(p_1^2\frac{u^k_{i+1j} - 2u^k_{i,j} + u^k_{i-1j}}{h_x^2} + p_2^2\frac{u^k_{ij+1} - 2u^k_{i,j} + u^k_{ij-1}}{h_y^2}\\
&+ 2p_1p_2\frac{u^k_{i+1j+1} + u^k_{i-1,j-1} - u^k_{i+1j-1} - u^k_{i-1j+1}}{4h_xh_y}\Big)\\
&+ 2\frac{\tau}{h_x}\Big(p_1^2\frac{u^k_{i+2j} - 2u^k_{i+1,j} + u^k_{ij}}{h_x^2} + p_2^2\frac{u^k_{i+1j+1} - 2u^k_{i+1,j} + u^k_{i+1j-1}}{h_y^2}\\
&+ 2p_1p_2\frac{u^k_{i+2j+1} + u^k_{i,j-1} - u^k_{i+j-1} - u^k_{ij+1}}{4h_xh_y}\Big) \qquad Eq.(11.2)\\
&= (1-2\frac{\tau}{h_x})(u_{pp})^k_{i,j} + 2\frac{\tau}{h_x}(u_{pp})^k_{i+1,j} \qquad (11.4)
\end{aligned}
$$

Since we assume that $\tau \leq \frac{h_x}{2}$, the following holds:

$$
(u_{pp})^{k+1}_{i,j} \leq (1-2\frac{\tau}{h_x})(u_{pp})^k_{i,j} + 2\frac{\tau}{h_x}(u_{pp})^k_{i+1,j} < 0 \qquad (11.5)
$$

Extending Shock Filters

Nonlinear isotropic diffusion can be reformulated as follows:

$$
\begin{aligned}
u_t &= \operatorname{div}(g(|\boldsymbol{\nabla} u|^2)\boldsymbol{\nabla} u)\\
&= g(|\boldsymbol{\nabla} u|^2)\operatorname{div}(\boldsymbol{\nabla} u) + g'(|\boldsymbol{\nabla} u|^2)2\big(u_x(u_xu_{xx} + u_yu_{xy}) + u_y(u_yu_{yy} + u_xu_{xy})\big)\\
&= g(|\boldsymbol{\nabla} u|^2)\Delta u + 2g'(|\boldsymbol{\nabla} u|^2)(u_x^2u_{xx} + 2u_xu_yu_{xy} + u_y^2u_{yy})\\
&= g(|\boldsymbol{\nabla} u|^2)\Delta u + 2g'(|\boldsymbol{\nabla} u|^2)\boldsymbol{\nabla} u^T\mathcal{H}(u)\boldsymbol{\nabla} u\\
&= g(|\boldsymbol{\nabla} u|^2)\Delta u + 2g'(|\boldsymbol{\nabla} u|^2)|\boldsymbol{\nabla} u|u_{\eta\eta}|\boldsymbol{\nabla} u| \qquad (11.6)
\end{aligned}
$$

Bibliography

[1] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel. Axioms and fundamental equations in image processing. *Archive for Rational Mechanics and Analysis*, 123:199–257, 1993.

[2] L. Alvarez and L. Mazorra. Signal and image restoration using shock filters and anisotropic diffusion. *SIAM Journal on Numerical Analysis*, 31:590–605, 1994.

[3] S. Bettahar and A. B. Stambouli. Shock filter coupled to curvature diffusion for image denoising and sharpening. *Image Vision Computing*, 26(11):1481–1489, November 2008.

[4] F. Bornemann and T. März. Fast image inpainting based on coherence transport. *Journal of Mathematical Imaging and Vision*, 28(3):259–278, July 2007.

[5] P. Peter C. Schmaltz. Image compression. Lecture Notes, 2020. Mathematical Image Analysis Group, Saarland University, Saarbrücken.

[6] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[7] 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*, 32:1895–1909, 1992.

[8] 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, 1997.

[9] J. H. Elder. Are edges incomplete? *International Journal of Computer Vision*, 34(2/3):97–122, 1999.

[10] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, Interlaken, Switzerland, June 1987.

[11] S. Fu, Q. Ruan, W. Wang, and J. Chen. Region-based shock-diffusion equation for adaptive image enhancement. In Nanning Zheng, Xiaoyi Jiang, and Xuguang Lan, editors, *Advances in Machine Vision, Image Processing, and Pattern Analysis*, pages 387–395, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[12] G. Gilboa, N. A. Sochen, and Y. Y. Zeevi. Regularized shock filters and complex diffusion. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Computer Vision – ECCV 2002*, volume 2350 of *Lecture Notes in Computer Science*, pages 399–413. Springer, Berlin, 2002.

[13] R. M. Haralick. Digital step edges from zero crossing of second directional derivatives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(1):58–68, 1984.

[14] R. Hummel and R. Moniot. Reconstructions from zero-crossings in scale space. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37:2111–2130, 1989.

[15] P. Kornprobst, R. Deriche, and G. Aubert. Image coupling, restoration and enhancement via PDEs. In *Proc. 1997 IEEE International Conference on Image Processing*, volume 4, pages 458–461, Washington, DC, October 1997.

[16] H. P. Kramer and J. B. Bruckner. Iterations of a non-linear transformation for enhancement of digital images. *Pattern Recognition*, 7:53–58, 1975.

[17] I. T. Young L. J. van Vliet and G. L. Beckers. A nonlinear Laplace operator as edge detector in noisy images. *Computer Vision, Graphics, and Image Processing*, 45(2):167–195, 1989.

[18] T. Lindeberg. Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:234–254, 1990.

[19] P. Maragos. Morphological filtering for image enhancement and feature detection. In A. C. Bovik, editor, *Image and Video Processing Handbook*, chapter 3.3, pages 135–156. Elsevier, Orlando, 2005.

[20] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London, Series B*, 207:187–217, 1980.

[21] S. Osher and L. I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis*, 27:919–940, 1990.

[22] 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.

[23] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM Journal on Numerical Analysis*, 29:867–884, 1992.

[24] G. Sapiro, R. Kimmel, D. Shaked, B. B. Kimia, and A. M. Bruckstein. Implementing continuous-scale morphology via curve evolution. *Pattern Recognition*, 26:1363–1372, 1993.

[25] K. Schaefer. Morphological differentiation. Bachelor's thesis, Saarland University, Saarbrücken, 2019.

[26] J. G. M. Schavemaker, M. J. T. Reinders, J. J. Gerbrands, and E. Backer. Image sharpening by morphological filtering. *Pattern Recognition*, 33:997–1012, 2000.

[27] C. Schmaltz, P. Peter, M. Mainberger, F. Ebel, J. Weickert, and A. Bruhn. Understanding, optimising, and extending data compression with anisotropic diffusion. *International Journal of Computer Vision*, 108(3):222–240, July 2014.

[28] J. Serra and G. Matheron. The birth of mathematical morphology, June 2000. ISMM 2000, Xerox Center Palo-Alto.

[29] J. Weickert. Theoretical foundations of anisotropic diffusion in image processing. *Computing Supplement*, 11:221–236, 1996.

[30] J. Weickert. *Anisotropic Diffusion in Image Processing.* Teubner, Stuttgart, 1998.

[31] J. Weickert. Fast segmentation methods based on partial differential equations and the watershed transformation. In P. Levi, R.-J. Ahlers, F. May, and M. Schanz, editors, *Mustererkennung 1998*, pages 93–100. Springer, Berlin, 1998.

[32] 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.

[33] J. Weickert. Image processing and computer vision. Lecture Notes, 2018. Mathematical Image Analysis Group, Saarland University, Saarbrücken.

[34] J. Weickert. Sparse inpainting with anisotropic integrodifferential operators. SIAM Conference on Imaging Sciences, 2018. Mathematical Image Analysis Group, Saarland University, Saarbrücken.

[35] J. Weickert. Differential equations in image processing and computer vision. Lecture Notes, 2019. Mathematical Image Analysis Group, Saarland University, Saarbrücken.

[36] J. Weickert. Personal communication, 2020. Mathematical Image Analysis Group, Saarland University.

[37] M. Welk, J. Weickert, and I. Galić. Theoretical foundations for spatially discrete 1-D shock filtering. *Image and Vision Computing*, 25(4):455–463, 2007.

[38] J. Xiao, G. Pang, Y. Zhang, Y. Kuang, Y. Yan, and Y. Wang. Adaptive shock filter for image super-resolution and enhancement. *Journal of Visual Communication and Image Representation*, 40:168 – 177, 2016.

[39] A. L. Yuille and T. A. Poggio. Scaling theorems for zero crossings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):15–25, January 1986.

List of Figures