Nonlinear Filtering of Video Sequences Using Contraction Theory

by

Martin A. Grepl

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2001

Author .....
Department of Mechanical Engineering
May 11, 2001

Certified by .....
J.-J. E. Slotine
Professor of Mechanical Engineering and Information Science,
Professor of Brain and Cognitive Science
Thesis Supervisor

Accepted by .....
Ain A. Sonin
Chairman

Nonlinear Filtering of Video Sequences Using Contraction Theory

by

Martin A. Grepl

Submitted to the Department of Mechanical Engineering
on May 11, 2001, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

The focus of this work is the application of partial differential equations (PDE's) to image and image sequence processing with emphasis on nonlinear PDE's. The main goal of the nonlinear filtering of images using PDE's is to achieve simultaneous noise removal and edge enhancement. First, the process leading to PDE's in image processing is reviewed and existing methods are classified into three basic types: nonlinear diffusion and its variational formulation, curvature flows, and shock filters. The methods are analyzed; their merits and drawbacks are discussed. Next, a result obtained from extending contraction analysis to the stability analysis of a certain class of nonlinear reaction-diffusion equations is presented. The result guarantees global stability and exponential convergence of the systems under consideration. The results obtained for static image enhancement are then extended to time-varying image sequences. A new approach for the enhancement of image sequences with time-varying uncorrelated noise is presented based on the above results. We propose an algorithm which is based on the idea of alternating direction implicit (ADI) methods in numerical analysis to solve a three-dimensional minimization problem. The algorithm is fast, recursive and may be used for real-time applications. Some properties of the new scheme are analyzed and its performance is shown on synthetic data.

Thesis Supervisor: J.-J. E. Slotine
Title: Professor of Mechanical Engineering and Information Science, Professor of Brain and Cognitive Science

Acknowledgments

First of all I want to express my deepest gratitude to my advisor, Professor J.-J. E. Slotine, for giving me the opportunity to work on this interesting field. He was a constant source of support and ideas.

I am most grateful to my family in Germany who always supported and encouraged my work.

I furthermore deeply appreciate all the friendships I found here and who made my life thoroughly enjoyable. I especially want to thank Karen for her love and support.

I am indebted to the University of Stuttgart for giving me the opportunity to spend my exchange program at MIT and to the 'Studienstiftung des deutschen Volkes' and the 'Dr. Jürgen Ulderup Stiftung' for their financial support during my first year of study at MIT.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

Image enhancement finds applications in such a wide variety of fields. In the medical field for instance, diagnosis often relies on information taken from MRI (Magnetic Resonance Imaging) or sonography. In aeronautics, autonomous landing and ATR (Automatic Target Recognition) require the use of radar images. It is common to all of those images that they are generally very noisy. To be able to get information out of such noisy images, it is necessary to remove noise from the image without sacrificing the useful detail and to enhance or highlight certain features such as edges or boundaries.

In the last decade there has been an increased interest in the use of partial differential equations (PDE) in image enhancement to achieve these goals. In this context the noisy image is taken as the initial condition for a time-dependent PDE. The restored image is either given by the steady-state solution of the equation or by stopping the evolution at a certain time. Very good results have been reported for the enhancement of a single image, while the enhancement of time-varying image sequences or movies has not found as much attention in the past.

In this thesis we will review the process leading to PDE's in image processing and show three main approaches. The first one can be classified as nonlinear diffusion and is closely related to the heat or diffusion equation. The diffusion equation was the

first PDE used in image processing because of its relation to low-pass filtering using Gaussian convolution. In the original approach the diffusion term was modified to stop the diffusion at edges, thus obtaining a nonlinear diffusion. However, the same results can be reached using a variational formulation, as will be shown in the review. The second approach makes use of curvature flows by interpreting the image as a collection of iso-intensity contours that can be evolved. The flow by curvature is also known as the geometric heat equation. The last approach is shock filters, which are related to results known from shock wave calculations in fluid dynamics.

The goal of this thesis is to extend the results from two-dimensional static images to the enhancement of three-dimensional image sequences, and to apply state-of-the-art methods in nonlinear stability analysis known as *contraction analysis* to image processing. The main advantage of this new method is that global exponential convergence rates can be quantified, and it suggests specific choices of nonlinearities and image coupling terms.

Finally, we will propose a new scheme for real-time image sequence processing. It is based on the idea of alternating direction implicit (ADI) methods in numerical analysis. Results on synthetic data and some properties of this fast and recursive algorithm will be given.

1.2 Concepts and Notation

Before reviewing PDE's in image processing the following concepts and notations have to be introduced. We define an image as an intensity function $u(x, y)$ on a domain $\Omega \subset \mathbb{R}^2$. In general, $u(x, y) : \mathbb{R}^2 \to \mathbb{R}^+$ represents a gray scale image, where $u(x, y)$ is the gray scale value and (x, y) specifies spatial position, i.e., $u(x_0, y_0)$ is an absolute light intensity at the point (x_0, y_0) in the image plane. Thus we can write

$$u = u(x, y) \qquad x, y \in \Omega.$$

The gray-level can take on values in the range 0 to 255. However, in this thesis this range is renormalized to (0,1). Here we employ the standard convention that 0 corresponds to black and 1 corresponds to white.

The noise we are dealing with is modeled as additive Gaussian white noise with mean μ and variance σ. The model we use for the noisy image is given by

$$u_0(x,y) = u(x,y) + n(x,y), \tag{1.1}$$

where $n(x,y)$ denotes the noise, $u(x,y)$ is the clean image, and $u_0(x,y)$ is the observed noisy intensity function. Figure 1-1 shows a clean image $u(x,y)$ on the left and the noisy test image $u_0(x,y)$ on the right. The square has a brightness value of 0.25 while the one of the background is 0.75. We will use this synthetic image to compare the performance of the enhancement schemes we will discuss. The noise added to the clean image has mean $\mu = 0$ and variance $\sigma = 0.1$ throughout this work.

In image sequences (videos) the image intensity changes over time t. Now, the intensity u is also a function of time, $u = u(x,y,t)$, and the image sequence is defined by $u(x,y,t) : \Omega^+ \subset \mathbb{R}^3$, where we use the superscript '+' to denote the additional dependence on time. We will consider two different models of image sequences. In the first one the noise is assumed to be time-invariant, leading to the equation

$$u_0(x,y,t) = u(x,y,t) + n(x,y). \tag{1.2}$$

In the second case the noise is taken to be time-varying and uncorrelated in time giving

$$u_0(x,y,t) = u(x,y,t) + n(x,y,t). \tag{1.3}$$

Based on those models we can state the main goal of image enhancement: *Find the "best" estimate $\hat{u}(x,y)$ of the clean image $u(x,y)$ from the given noisy information $u_0(x,y)$.* To measure the performance of the different enhancement schemes, we have to define what we mean by "best". This is a difficult task since everyone's perception of the results may be different. However, since we deal with noisy and restored images (signals) and the clean image is actually available, we can compute the *signal-to-noise-ratio* defined by

$$\mathrm{SNR}(u_1/u_2) = 10\ \log_{10}\left[\frac{\sigma^2(u_2)}{\sigma^2(u_1 - u_2)}\right]\ , \tag{1.4}$$

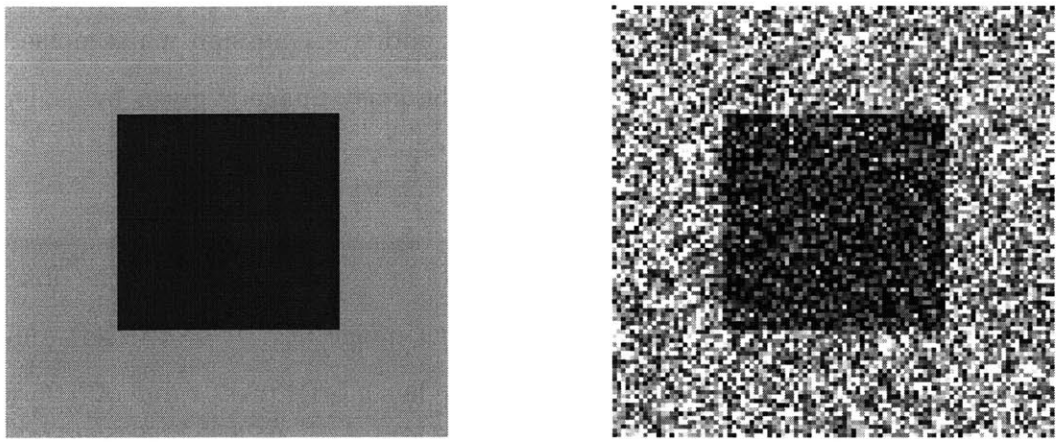

Figure 1-1: Synthetic image $u(x, y)$ on the left and test image $u_0(x, y)$ obtained by adding Gaussian white noise to $u(x, y)$.

where σ is the variance. To compare the results we will give values of SNR(u/u_0) and SNR$(u/\hat{u})$, where $\hat{u}$ is the estimate resulting from the enhancement schemes. Note that the higher the signal to noise ratio the better the restoration of the noisy image in a mathematical sense.

At this point we can define more accurately what we mean by an edge or boundary in the image. We say that an edge is simply a jump or discontinuity in the image intensity $u(x, y)$. Thus, $u(x, y)$ is singular at these points and the gradient of $u(x, y)$ is strong. We determine that points where this happens carry important information about the scene being observed. We now have two main tasks. On the one hand we wish to smooth the homogeneous regions of the image with two objectives: noise elimination and image interpretation. On the other hand, we wish to keep the accurate location of the boundaries of these regions. Unfortunately, these are two conflicting requirements. Furthermore, the classification of some information as "noise" and other information as "important" is not clear, and techniques to correspondingly filter images must, at some level, reflect this ambiguous decision. One natural filter exploits the idea of scale - one can try to remove information that occupies a small amount of the image, such as spots of non-matching brightness values or small oscillations in boundaries of objects.

As an example consider the observation of a forest. On the coarsest scale we can only determine the forest as a huge green spot. Decreasing the scale will enable us to distinguish the different treetops. And on the finest scale we can even recognize the individual leaves. Returning to the noisy image we can state that the fine or inner scale corresponds to noise while the coarse or outer scale is used to describe the main features. It is therefore important to look at the image in several scales. The goal of image enhancement is to diffuse the inner scale while keeping the outer scale. For the previous example this would mean that we want the leaves to merge together before the different treetops merge.

The main reason for diffusing images is that diffusion naturally produces a "scale-space", e.g., a fine-to-coarse family of derived images. The parameter we will usually use to encode the scale or resolution is the time t, and coarse/fine scale means low/high resolution respectively. Hence, looking at the image at different scales means different stages of filtering. And going from fine to coarse scales it is desirable to keep important image features but suppress artifacts like noise.

The algorithms that we will discuss are based on the formulation of partial differential equations. The steady-state solution $\hat{u}(x, y, t)$ of the differential equation gives the restored image. Using the original noisy image $u_0(x, y)$ as the initial condition, where $u(x, y) : \mathbb{R}^2 \to \mathbb{R}^+$ is an image in the continuous domain, and t is a time parameter, the image processing methods can be put in the form

$$\frac{\partial \hat{u}}{\partial t} = \Psi(\hat{u}, \nabla\hat{u}, \Delta\hat{u}, u_0) \tag{1.5}$$

$$\frac{\partial \hat{u}}{\partial n} = 0 \quad \text{on } \partial\Omega \tag{1.6}$$

$$\hat{u}(x, y, t = 0) = u_0(x, y). \tag{1.7}$$

Thus, we will have zero gradient boundary conditions for all evolution procedures and Ψ can be any function (linear or nonlinear) of its arguments.

For convenience we will denote partial derivatives with subscripts in the sequel, that is

$$\frac{\partial u(x, y, t)}{\partial t} = u_t(x, y, t) \quad \text{and} \quad \frac{\partial u(x, y, t)}{\partial x} = u_x(x, y, t).$$

The same notation holds for derivatives with respect to y and higher derivatives,

e.g., we will write u_{xx} for the second derivative w.r.t. x. The ∇-operator and the Laplacian Δ are defined by

$$\nabla = \left(\frac{\partial}{\partial x} , \frac{\partial}{\partial y} \right) \quad \text{and} \quad \Delta = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} ,$$

respectively.

It is possible to rewrite equation (1.5) in a different form by using $\eta - \xi$ coordinates. Here, η denotes the direction of the gradient and ξ the direction perpendicular to the gradient, also known as the direction of level sets [30]

$$\eta = \frac{[u_x, u_y]}{\sqrt{u_x^2 + u_y^2}}, \qquad \xi = \frac{[-u_y, u_x]}{\sqrt{u_x^2 + u_y^2}}. \tag{1.8}$$

The second-order derivatives in these directions are related to the derivatives in the x and y direction by

$$u_{\eta\eta} = \frac{u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy}}{u_x^2 + u_y^2} \tag{1.9}$$

$$u_{\xi\xi} = \frac{u_y^2 u_{xx} - 2u_x u_y u_{xy} + u_x^2 u_{yy}}{u_x^2 + u_y^2}, \tag{1.10}$$

where an equivalent notation for equation (1.10) is given by

$$u_{\xi\xi} = |\nabla u| \ \text{div} \left(\frac{\nabla u}{|\nabla u|} \right). \tag{1.11}$$

After the coordinate change equation (1.5) can be written as

$$\hat{u}_t = c_\eta \hat{u}_{\eta\eta} + c_\xi \hat{u}_{\xi\xi} + f(\hat{u}, u_0), \tag{1.12}$$

where c_η and c_ξ are functions involving derivatives lower than second order and $f(\hat{u}, u_0)$ is a function involving no derivatives, e.g., a feedback term $(u_0 - \hat{u})$. Note that $\hat{u}_{\eta\eta}$ is a diffusion along the direction of the gradient while $\hat{u}_{\xi\xi}$ diffuses in the direction perpendicular to the gradient. Thus c_η controls the diffusion across the edge and c_ξ controls the diffusion parallel to the edge. Bearing this fact in mind we can state the conditions c_η and c_ξ should meet for the enhancement procedure to perform "best". Since we want to have diffusion inside regions but not across edges, the conditions to obtain no smoothing of edges while removing noise within uniform regions are

$$\lim_{|\nabla \hat{u}| \to 0} c_\eta = \lim_{|\nabla \hat{u}| \to 0} c_\xi = \alpha > 0, \tag{1.13}$$

$$\lim_{|\nabla \hat{u}| \to \infty} c_\eta = 0 \ \text{ and } \lim_{|\nabla \hat{u}| \to \infty} c_\xi = \beta > 0. \tag{1.14}$$

As introduced above we used the assumption here that low gradients correspond to noise while high gradients correspond to important features like edges and boundaries.

1.3 Thesis Organization

Chapters 2 through 3 present the three main approaches to image processing which use partial differential equations.

Chapter 2 first shows how filtering by Gaussian convolution leads to the first PDE in image processing, namely the heat or diffusion equation. Starting from this idea a modification to the diffusion equation is introduced that uses nonlinearities in the divergence operator to achieve a better performance. In the second part of this chapter we present a variational approach whose solution also leads to an enhancement scheme using nonlinear diffusion equations.

Chapter 3 describes the other two approaches. The first one are curvature flows, in whose context the image is interpreted as a collection of curves, strictly speaking the iso-intensity contours of the image, which can be evolved. The second one is related to shock filters and applies ideas that stem from fluid dynamics.

A result from contraction analysis and its extension to the stability analysis of reaction-diffusion equations is stated in the beginning of Chapter 4. Afterwards the result is applied to one- and two-dimensional signal and image enhancment. The main advantage of contraction analysis is that we can guarantee the global stability and determine the exponential convergence rate of the enhancement scheme we are going to discuss.

Finally, Chapter 5 deals with the enhancement of image sequences. First, an extension of the enhancement algorithms for static images to time-varying images will be given. It turns out that this approach is restricted only to certain models of the image sequence. Then, a new idea for image sequence enhancement is introduced and a fast and recursive scheme will be derived.

A summary of the main ideas and their merits follow in Chapter 6. Furthermore suggestions for future work will be given.

Since the topic of image enhancement is vast its treatment in this thesis is far from complete. For more general information the interested reader is referred to [1, 8, 10, 27, 39, 40] and the references therein. Specific information on nonlinear scale-spaces can be found in [10, 11, 12], on affine versions of scale-space in [35, 36, 37], and on morphological scale-space in [41, 42].

Chapter 2

Nonlinear Diffusion and its Variational Formulation

2.1 Introduction

This chapter outlines the introduction of partial differential equations in image enhancement. In the first part the original approach is presented. It is based on the connection between Gaussian convolution and the first PDE used in this field: the diffusion equation. We will also show that a substantially better performance can be achieved by using nonlinearities in the divergence term, leading to the notion of nonlinear diffusion.

In the second part we will see that nonlinear diffusion equations can also be derived using a variational formulation. A PDE is obtained by solving the Euler-Lagrange equations associated to the minimization problem using a steepest descent method. Several examples for the functional to be minimized that are proposed in the literature are presented.

2.2 The Gaussian Scale-Space

2.2.1 Gaussian Convolution

In the classical theory, noise elimination and image enhancement are achieved by a low pass filtering. The idea is to eliminate spurious details with high frequency content like noise while maintaining the strong image features with low frequencies such as edges. The main edges of the original image can then be identified at places where the gradient of the filtered image has an extremum. Denoting the filtered image by $\hat{u}(x, y)$, the edges are curves where $|\nabla \hat{u}(x, y)|$ is a maximum. Thus, finding edges means finding extrema in the gradient of $\hat{u}(x, y)$. A maximum of the gradient on the other hand is equivalent to the condition that the Laplacian of the smoothed signal $\Delta \hat{u}(x, y)$ changes sign. Points in the image where this happens are also called *zero-crossings* of the Laplacian. Marr and Hildreth who developed this theory in [27] speculated that zero-crossings which spatially coincide over several *scales* are "physically significant."

These ideas have been improved and formalized by Witkin [43] and Koenderink [21]. In the classical theory noisy signals are usually filtered by convolving them with a Gaussian function of a certain variance σ. The degree of blurring is controlled by the variance of the Gaussian function. A larger variance means a higher rate of smoothing; thus σ controls the scale of the smoothed signal. Witkin suggests that the signal be observed at several scales and to determine how the signal evolves as the scale is varied (say, from fine to coarse). This idea leads to the notion of *scale-space* in signal processing. Witkin proposed to filter the original signal $u_0(x)$ by convolving it with Gaussian functions of increasing variance σ

$$u(x, \sigma) = u_0(x) * G_\sigma(x) = \int_{-\infty}^{+\infty} u_0(y) \, \frac{1}{\sigma\sqrt{2\pi}} \, e^{-\frac{(x-y)^2}{2\sigma^2}} \, dy$$

where "$*$" denotes convolution with respect to x. By continuously varying the scale parameter σ the original signal $u_0(x)$ is embedded in a family of derived signals $u(x, \sigma)$. The $x - \sigma$ plane is called the scale-space and the function $u(x, \sigma)$ defines a surface on the $x - \sigma$ plane, called the *scale-space image* of $u_0(x)$ [43]. Figure 2-1 shows a family of one-dimensional signals obtained by convolving the original signal at the bottom

with a Gaussian of increasing variance (from bottom to top). Each curve represents a cross-section of the scale-space image for a certain σ = constant. Note that the larger the value of σ the higher the degree of blurring. Also note that the high frequency noise is removed very fast, while the important features with lower frequencies in the signal remain more stable across the scales before they also disappear. The blurring of important features is one of the problems associated with Gaussian smoothing of signals and images. Another problem becomes apparent if we try to find the main edges of the original signal $u_0(x)$.

Following the classical theory introduced above we have to look for maxima in the gradient of the smoothed signal $u_x(x, \sigma)$ for a certain $\sigma > 0$, or equivalently for zero-crossings of the one-dimensional Laplacian $u_{xx}(x, \sigma)$. By varying σ continuously again we can now interpret $u_{xx}(x, \sigma)$ as another surface in the $x - \sigma$ plane. To obtain the zero-crossings of $u_{xx}(x, \sigma)$ we can simply plot the contours of $u_{xx}(x, \sigma) = 0$ in the scale-space. Figure 2-2 shows the same signal we used in Figure 2-1 on the bottom and the contours of $u_{xx}(x, \sigma) = 0$ in the scale-space.

We can thus see how the zero-crossings or edges of our original signal evolve over scale. Let us point out the two main features of Gaussian filtering:

- The contours of small oscillations emanating from noise disappear first, while the ones from strong image features remain more stable across the scales. We already mentioned this property in the context of Figure 2-1. For the largest σ in Figure 2-2 only two zero-crossings remain. These zero-crossings correspond to the coarsest signal (the one on the top) of Figure 2-1. By following the contours of the strong features from fine to coarse scales one can observe the other main problem. The contours are twisted and thus the zero-crossings of the smoothed signal give an inexact account for the location of the true edges of our original image. Both problems, blurring and shifting of edges happen because the Gaussian function is an isotropic operator. That is, since it smooths in all directions, sharp boundaries will also be blurred.

- The other main point to observe is that the contours in the scale-space are

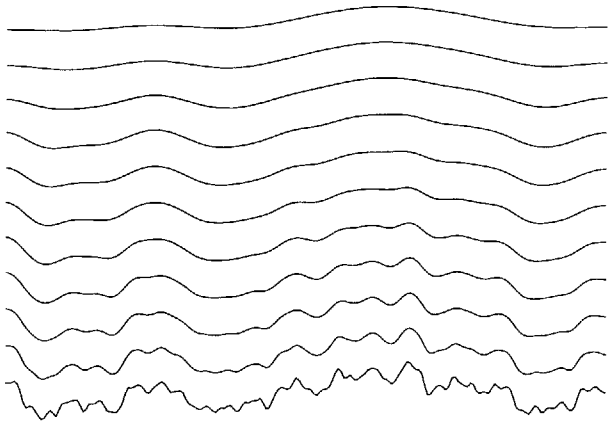

Figure 2-1: A family of one-dimensional signals $u(x, \sigma)$ obtained by convolving the original one (bottom) with Gaussian functions of increasing variance (bottom to top).

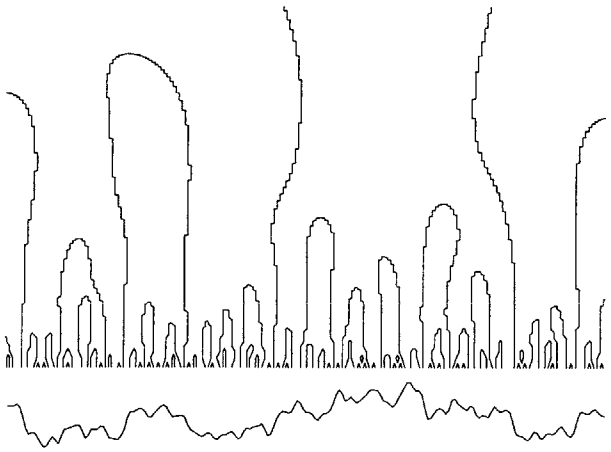

Figure 2-2: Original signal (bottom) and contours of $u_{xx}(x, \sigma) = 0$ in scale-space. The horizontal axis corresponds to x, and the vertical axis to σ with the coarsest scale on top.

closed above but never below. This is the most desirable property when using Gaussian convolution. To understand what it means recall that every contour corresponds to a zero-crossing of the Laplacian of the original signal. Therefore moving from fine to coarse scale, zero-crossings are never created as the scale increases. An equivalent statement is that no new extrema, and thus edges, should appear while increasing the scale. It is therefore possible to identify the main edges at a coarse scale and then follow them backward by making the scale decrease. This method could theoretically give the exact location of the edges. However, the implementation is rather involved and is unstable because of the appearance of new edges across scales and the multiple thresholdings involved in the edge detection at each scale.

The second property just mentioned is also known as the causality requirement[1] and was introduced by Koenderink [21]. Starting from the condition that the causality requirement has to hold for every scale-space evolution, Koenderink proposed the first partial differential equation in image processing.

2.2.2 The Diffusion Equation

Koenderink [21] noticed that the convolution of a signal with a Gaussian is equivalent to the solution of the diffusion equation, with the signal as initial condition. If this initial condition is denoted by $u_0(x, y)$, the "scale-space" analysis associated with $u_0(x, y)$ consists in solving the system

$$\begin{aligned} \hat{u}_t(x, y, t) &= \operatorname{div}(c\ \nabla\hat{u}(x, y, t)) = c\ \Delta\hat{u}(x, y, t) \\ \hat{u}(x, y, 0) &= u_0(x, y), \end{aligned} \tag{2.1}$$

[1]Babaud *et al* [5] obtained the result that the Gaussian filter is the unique linear operator for which the causality criterion holds.

where c is the diffusion constant[2]. The solution of equation (2.1) for $c = 1$ is $\hat{u}(x, y, t) = G_t(x, y) * u_0(x, y)$, where

$$G_t(x, y) = \frac{1}{\sqrt{4\pi t}} e^{-\frac{x^2+y^2}{4t}} \tag{2.2}$$

is a Gaussian function with time playing the role of the variance, related by $\sigma = \sqrt{2t}$. We can now check what happens to the test image from Figure 1-1 over time by using it as the initial condition for the diffusion equation. We know that the result is equivalent to convolving it with a two-dimensional Gaussian function. Thus the noise will be removed, but also the edges will be blurred and might be shifted from their true location. This is illustrated in Figure 2-3. It shows the input image $u_0(x, y)$ on the left which has a signal to noise ratio of $\mathrm{SNR}(u/u_0) = 3.4$ and the result on the right with $\mathrm{SNR}(u/\hat{u}) = 8.8$. At the core of those problems is the linearity of the Laplacian operator. It produces a displacement or loss of the edges that are present in the image. This makes it necessary to find nonlinear operators as we will see in the next section.

2.3 Anisotropic Diffusion

An important improvement of the classical scale-space analysis stems from Perona and Malik [31]. Perona and Malik suggested to introduce part of the edge detection in the filtering itself, thus ensuring that the edges remain much more stable across the scales. The idea is to choose the diffusion constant c of equation (2.1) such that it is conditional. The aim is to encourage smoothing within a region in preference to smoothing across boundaries. This corresponds to choosing $c = 1$ in the interior of each region and $c = 0$ across boundaries. In contrast to isotropic smoothing the blurring would only take place inside uniform regions and the boundaries would remain sharp, thus the notion of anisotropic diffusion.

But how do we know if we are inside a region or on a boundary? Let us assume that a low gradient in the image corresponds to noise while a high gradient corresponds to

[2]For the sake of clarity we will omit the arguments x, y, and t in the following equations and only mention them in the initial conditions.

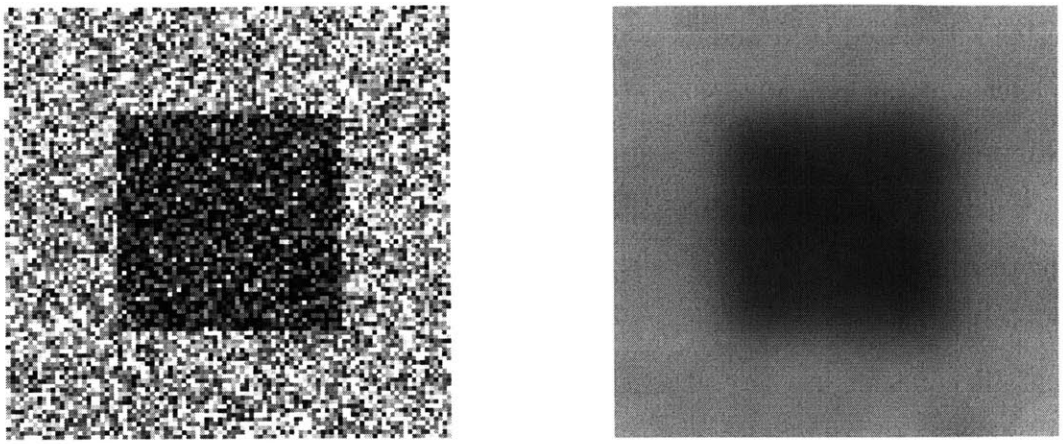

Figure 2-3: Original image on the left and the filtered image produced by the diffusion equation.

edges. Then we can take c as a function of $|\nabla\hat{u}|$, that is $c = g(|\nabla\hat{u}|)$. According to the above-mentioned strategy $g(|\nabla\hat{u}|)$ has to be a nonnegative monotonically decreasing function with $g(0) = 1$ and $g(|\nabla\hat{u}|) \to 0$ as $|\nabla\hat{u}|$ tends to infinity. Following Perona and Malik the model can be written as

$$\begin{aligned} \hat{u}_t &= \operatorname{div}(g(|\nabla\hat{u}|)\ \nabla\hat{u}) \\ \hat{u}(x,y,0) &= u_0(x,y), \end{aligned} \tag{2.3}$$

where $g(|\nabla\hat{u}|)$ is a positive non increasing function given by

$$g(s) = e^{-(\frac{s}{K})^{\alpha}} \qquad (\alpha \geq 1) \tag{2.4}$$

or

$$g(s) = \frac{1}{1 + (\frac{s}{K})^{1+\alpha}} \qquad (\alpha \geq 1), \tag{2.5}$$

where K is a positive constant. On the left of Figure 2-4 the qualitative shape of $g(s)$ given by equation (2.4) is plotted. The diffusion will be lowered if $|\nabla\hat{u}(x,y)|$ is large to keep the exact location of edges. If $|\nabla\hat{u}(x,y)|$ is small the diffusion will tend to smooth still more around (x,y), giving the desirable result of blurring small discontinuities like noise and sharpening edges. By choosing different values for α and

K we are able to change the shape of $g(s)$. We can expect that there exists a certain threshold θ related to α and K which is crucial for the diffusion of the scheme. We will have a closer look at this point in Section 2.3.1.

Figure 2-5 shows the result of applying the Perona-Malik equation (2.3) to the test image, with $g(s)$ from equation (2.4), $\alpha = 1$, and $K = 0.2$. The signal to noise ratio of the test image is $\text{SNR}(u/u_0) = 3.4$ while that of the result is $\text{SNR}(u/\hat{u}) = 17.7$. We observe that the edges are better preserved than in Figure 2-3. However, the Perona and Malik equation has several practical and theoretical difficulties. These are shown in the next section.

2.3.1 Instabilities of the Perona-Malik Equation

The first problem that appears for most enhancement schemes is the stopping time. If the evolution procedure is not stopped all information would eventually be removed from the image. To obtain a good result it is therefore necessary to choose a stopping time in advance, which is not possible for machine applications.

The second problem comes from the nonlinearity inside the divergence operator. Let us have a closer look at equation (2.3) and rewrite its divergence term in the form of equation (1.12). Recall that η denotes the direction of the gradient and ξ the direction perpendicular to the gradient. In $\eta-\xi$ coordinates we obtain

$$\text{div}(g(|\nabla\hat{u}|)\nabla\hat{u}) = g(|\nabla\hat{u}|)\ \hat{u}_{\xi\xi} + h'(|\nabla\hat{u}|)\ \hat{u}_{\eta\eta}, \qquad \text{where } h(s) = s \cdot g(s). \tag{2.6}$$

This means that the diffusion perpendicular to the gradient is controlled by $g(s)$ while the diffusion along the gradient is controlled by $h'(s)$. The right-hand plot of Figure 2-4 shows the qualitative shape of $h(s)$. We can see that there exists a certain threshold value θ defined by $h'(\theta) = 0$ – the threshold we already mentioned in the last section. And since $h(s) = s \cdot g(s)$ the relation to α and K is straightforward. We can also observe from the plot that the function $h(s)$ is monotonically increasing for $s \leq \theta$ and monotonically decreasing beyond θ. We can thus distinguish three main properties of equation (2.6):

- For $h' > 0$ the equation behaves locally as a diffusion in both directions.

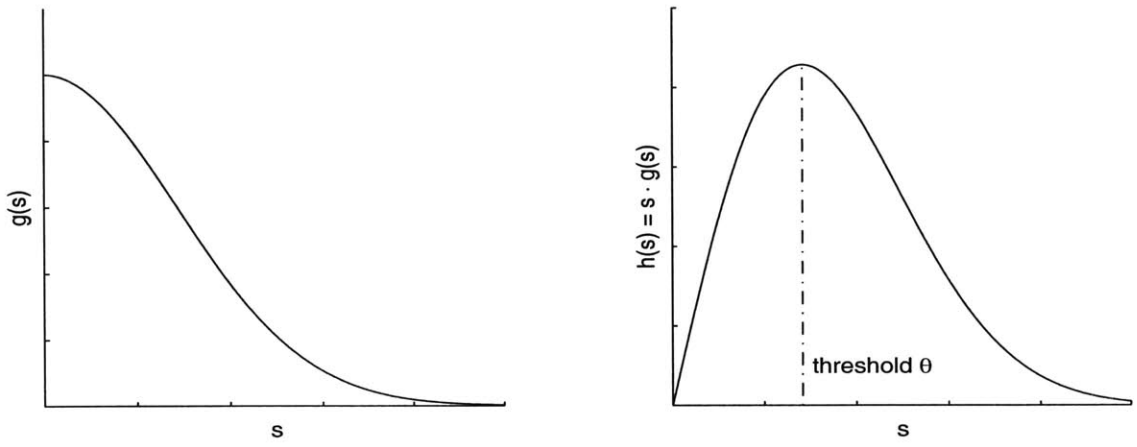

Figure 2-4: Left: Stopping function $g(s)$. Right: Function $h(s) = s \cdot g(s)$ inside the divergence operator.

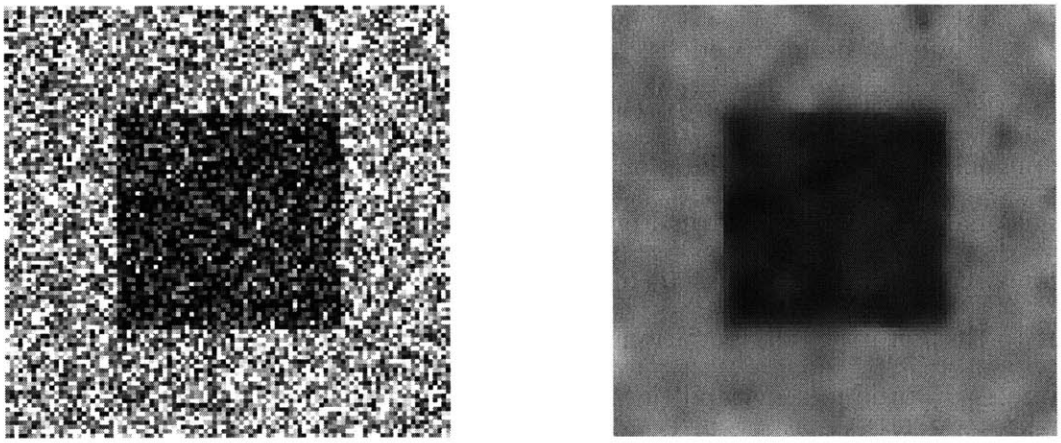

Figure 2-5: Original image on the left and solution of the Perona-Malik anisotropic diffusion equation.

- For $h' = 0$ the equation diffuses only in the direction perpendicular to the gradient.

- For $h' < 0$ we obtain a local inversion of the heat equation because of the negative speed of the diffusion in the direction of the gradient. The inverse diffusion equation is well known to be unstable and we cannot expect a meaningful solution.

Let us now have a look at the last point by considering the one-dimensional version of equation (2.3)

$$\hat{u}_t = \frac{\partial}{\partial x} h(\hat{u}_x) = h'(\hat{u}_x) \cdot \hat{u}_{xx}, \tag{2.7}$$

where $h(s)$ is defined as above. Again, for $h'(\hat{u}_x) \leq 0$ this equation acts like an inverse diffusion equation. In practice, that means that very similar pictures could produce divergent solutions and therefore different edges. In the context of image processing this can lead to the following two problems.

The first problem comes from the discretization of the Perona-Malik equation. In contrast to the diffusion equation which blurs all edges in the image, the inverse form has the capability of deblurring. Thus for certain values of $|\nabla \hat{u}|$ we should be able to deblur or build up edges. Figure 2-6 shows a step edge, the blurred edge and the steady state solution of the one-dimensional Perona-Malik equation (2.7). The phenomenon we see in the plot is known as staircasing. The corresponding two-dimensional version is shown in Figure 2-7. The original synthetic image is blurred by convolving it with a Gaussian function. Next, the blurred image is taken as the input to the Perona-Malik equation. As we can see in the 2-D case the choice of the parameter K produces instabilities in the solution of the equation. The number of new edges depends strongly on the value of K.

The second problem is related to the noise statistics of the image. Assuming that the signal is very noisy, the noise introduces very large, in theory unbounded oscillations of the gradient $\nabla \hat{u}$. Thus, the conditional smoothing introduced by the model will not give good results, since all these noise edges will be kept. Figure 2-8

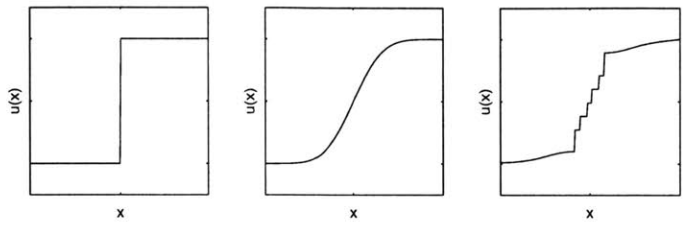

Figure 2-6: Application of Perona-Malik equation to blurred step edge. From left to right: original step edge, blurred version, steady state solution of equation (2.7).

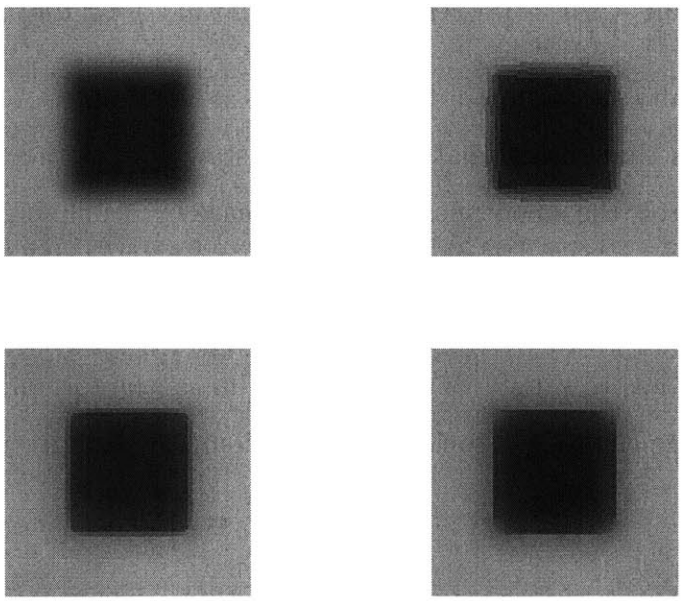

Figure 2-7: Application of Perona-Malik equation to a blurred image. From left to right, top to bottom: Original blurred image, application of Perona-Malik with $K = 0.03$, $K = 0.04$, and $K = 0.05$.

shows our test image on the left and the result of the Perona-Malik equation. The only difference to Figure 2-5 is that a smaller value for the constant K is chosen, thus lowering the threshold θ and the decision between smoothing and enhancement.

2.4 A Variational Formulation of Nonlinear Diffusion

In the last two sections we described the original ideas that introduced PDE's in image processing. We will now show that nonlinear diffusion equations for image enhancement can also be derived using a variational formulation [3]. Assuming that the model for the noisy image is given by equation (1.1), our goal is to find an estimate $\hat{u}(x,y)$ of the clean image $u(x,y)$ based on the noisy observation $u_0(x,y)$. The problem of finding this estimate can be formulated as the minimization problem:

$$\inf_{\hat{u}} \left\{ \iint_{\Omega} \left(\frac{1}{2} (\hat{u} - u_0)^2 + \phi(|\nabla \hat{u}|) \right) dx\, dy \right\}, \tag{2.8}$$

where ϕ is a penalty function still to be defined. While the first term in this energy functional assures that the estimate $\hat{u}(x,y)$ will stay close to the original image the second term controls the smoothness of the estimate by penalizing the gradient in a certain way. The penalty function ϕ controlling this smoothness is chosen to filter the noise within uniform regions and to simultaneously keep the main edges of the image. The conditions ϕ has to satisfy to achieve this aim can be formalized by looking at the Euler-Lagrange equation associated to equation (2.8) and given by

$$(\hat{u} - u_0) - \operatorname{div}\left(\frac{\phi'(|\nabla \hat{u}|)}{|\nabla \hat{u}|} \nabla \hat{u} \right) = 0. \tag{2.9}$$

Solving this equation by a steepest descent method with time as an artificial evolution parameter leads to the partial differential equation

$$\hat{u}_t = \operatorname{div}\left(\frac{\phi'(|\nabla \hat{u}|)}{|\nabla \hat{u}|} \nabla \hat{u} \right) + (u_0 - \hat{u}) = \operatorname{div}(g(|\nabla \hat{u}|)\, \nabla \hat{u}) + (u_0 - \hat{u}), \tag{2.10}$$

where Neumann boundary conditions are imposed on the boundary, the initial condition is given by $\hat{u}(x,y,t=0) = u_0(x,y)$, and $g(s)$ is defined by $g(s) = \frac{\phi'(s)}{s}$. Using

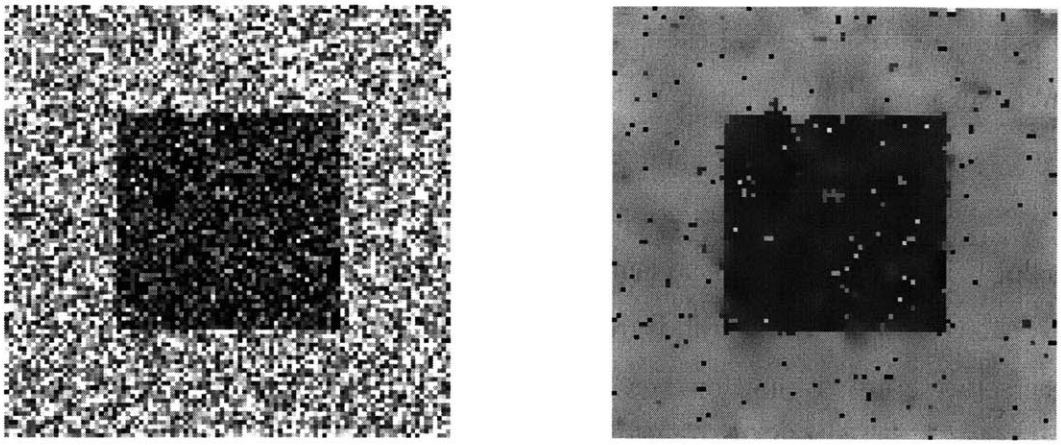

Figure 2-8: Original image on the left and solution of the Perona-Malik anisotropic diffusion for $K = 0.12$.

$\eta - \xi$ coordinates introduced in equation (1.8) the divergence term can be rewritten as

$$\operatorname{div}\left(\frac{\phi'(|\nabla\hat{u}|)}{|\nabla\hat{u}|}\,\nabla\hat{u}\right) = \frac{\phi'(|\nabla\hat{u}|)}{|\nabla\hat{u}|}\,\hat{u}_{\xi\xi} + \phi''(|\nabla\hat{u}|)\,\hat{u}_{\eta\eta}. \tag{2.11}$$

If we compare this formulation of the divergence term to equation (1.12) we can identify the diffusion coefficients c_η and c_ξ as

$$c_\eta = \phi''(|\nabla\hat{u}|) \quad \text{and} \quad c_\xi = \frac{\phi'(|\nabla\hat{u}|)}{|\nabla\hat{u}|}. \tag{2.12}$$

If we want a good restoration as described above c_η and c_ξ, and thus ϕ, have to verify the conditions stated in equation (1.13) and (1.14). Unfortunately the two conditions of equation (1.14) are incompatible for any function ϕ and cannot be satisified simultaneously. Condition (1.14) is therefore replaced by the following conditions imposed for high gradients [4, 8]

$$\lim_{|\nabla u|\to\infty} c_\eta = \lim_{|\nabla u|\to\infty} c_\xi = 0 \quad \text{and} \quad \lim_{|\nabla u|\to\infty} \left(\frac{c_\eta}{c_\xi}\right) = 0. \tag{2.13}$$

These conditions mean that the diffusion across the edge, governed by c_η, has to vanish more rapidly than the diffusion along the edge. This ensures that noise within uniform regions disappears before edges get blurred. It has been shown that for a

convex function $\phi(s)$ with linear growth at infinity the solution of equation (2.8) exists and is unique [3, 7]. Several choices for $\phi(s)$ are given in Table (2.1).

We can immediately see that the first two functions $\phi(s)$ in Table (2.1) are not convex and the existence and uniqueness of the solution is not guaranteed. Indeed, we already discussed both functions in Section 2.3 and saw that these choices lead to a local inversion of the heat equation in certain image regions. The remaining three choices are all convex and grow linearly at infinity; existence and uniqueness of the solution are given. However, only the last two choices additionaly satisify the conditions (1.13) and (2.13) stated for optimal restoration. Let us now have a closer look at three of the five choices given for $\phi(s)$.

2.4.1 Biased Anisotropic Diffusion

The second choice of ϕ has been proposed by Hebert *et al* [17] in the context of a generalized expectation-maximization algorithm and by Nordstroem [28] in the context of a minimization problem similar to the one presented above. The PDE obtained using the penalty function $\phi(s) = \frac{1}{2}\log(1+s^2)$ is given by

$$\begin{aligned} \hat{u}_t &= \operatorname{div}\left(\frac{\nabla\hat{u}}{1+(|\nabla\hat{u}|/K)^2}\right) + (u_0 - \hat{u}) \\ \hat{u}(x,y,0) &= u_0(x,y), \end{aligned} \tag{2.14}$$

where the constant K is introduced as a threshold or scaling parameter similar to section 2.3. Because of the feedback term $(u_0 - \hat{u})$ the equation has the advantage

Name of function $\phi(s)$	$\phi(s)$	$g(s) = \frac{\phi'(s)}{s}$
Perona-Malik [31]	$1 - \frac{1}{2}e^{-s^2}$	e^{-s^2}
Hebert-Leahy [17]	$\frac{1}{2}\log(1+s^2)$	$\frac{1}{1+s^2}$
Total Variation [33, 34]	s	$\frac{1}{s}$
Green [16]	$\log(\cosh(s))$	$\frac{\tanh(s)}{s}$
Hypersurfaces [3]	$\sqrt{1+s^2}-1$	$\frac{1}{\sqrt{1+s^2}}$

Table 2.1: Different choices of penalty function $\phi(s)$.

of having a nontrivial steady state, eliminating therefore the problem of choosing a stopping time. Despite this advantage however, the equation suffers from the same drawbacks that are mentioned in Section 2.3.1 for the Perona-Malik equation. Those drawbacks leading to the observed instabilities are originating from the non-convexity of $\phi(s)$.

Figure 2-9 shows the input image and the steady-state solution of equation (2.14) with $K = 0.2$. The original image has a signal to noise ratio of $\mathrm{SNR}(u/u_0) = 3.4$ and the result has $\mathrm{SNR}(u/\hat{u}) = 17.5$. Continued application of the equation produces no further change.

2.4.2 Total Variation Minimization

Rudin *et al* [33, 34] proposed to minimize the total variation of the image subject to constraints involving the statistics of the noise. The total variation is a measure of how oscillatory the image is and is defined by

$$\mathrm{TV}(u(x,y)) = \iint_{\Omega} |\nabla u(x,y)| \; dx \; dy. \tag{2.15}$$

The authors assume that the mean μ of the noise is zero and that the standard deviation σ is given. The constrained minimization problem can then be formulated in the following way:

$$\inf_{\hat{u}} \iint_{\Omega} |\nabla \hat{u}(x,y)| \;\; dx \; dy \tag{2.16}$$

subject to the constraints

$$\iint_{\Omega} \hat{u} \; dx \; dy \;\; = \;\; \iint_{\Omega} u_0 \; dx \; dy \tag{2.17}$$

$$\iint_{\Omega} \frac{1}{2}(\hat{u} - u_0)^2 \; dx \; dy \;\; = \;\; \sigma^2, \qquad \text{where } \sigma > 0 \text{ is given.} \tag{2.18}$$

The first constraint (2.17) is automatically satisfied since the noise has zero mean and the boundary conditions that will be used are $\frac{\partial \hat{u}}{\partial n} = 0$ on $\partial\Omega$. The second constraint (2.18) is enforced using Lagrange multipliers and leads to the minimization problem similar to the one given in equation (2.8)

$$\inf_{\hat{u}} \left\{ \iint_{\Omega} \left(\frac{\lambda}{2} (\hat{u} - u_0)^2 \; + \; |\nabla \hat{u}| \right) \; dx \; dy \right\}. \tag{2.19}$$

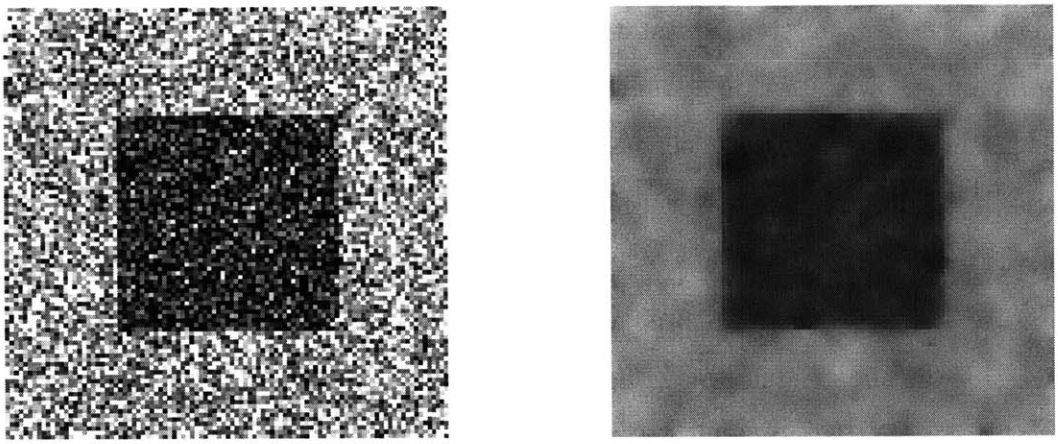

Figure 2-9: Original image on the left and the steady state solution of equation (2.14).

The PDE obtained from solving the Euler-Lagrange equation is therefore

$$\begin{aligned} \hat{u}_t &= \operatorname{div}\left(\frac{\nabla \hat{u}}{|\nabla \hat{u}|}\right) + \lambda(u_0 - \hat{u}), \\ \hat{u}(x, y, 0) &= u_0(x, y), \end{aligned} \tag{2.20}$$

where the Lagrange multiplier λ can be computed (since σ is assumed to be known) and is given by

$$\lambda = -\frac{1}{2\sigma^2} \iint_{\Omega} \left\{ |\nabla \hat{u}| - \left(\frac{(u_0)_x \hat{u}_x}{|\nabla \hat{u}|} + \frac{(u_0)_y \hat{u}_y}{|\nabla \hat{u}|} \right) \right\} dx\, dy. \tag{2.21}$$

Figure 2-10 shows the test image on the left with $\mathrm{SNR}(u/u_0) = 3.4$, and on the right the result obtained from the total variation minimization with $\mathrm{SNR}(u/\hat{u}) = 22.5$. The number of iterations until the steady-state is reached is 1000 and the time-step is $\Delta t = 5 \cdot 10^{-6}$. The reconstruction is very impressive and belongs to the best ones that can be achieved by the methods described in this work. Denoising algorithms based on minimization formulations for other models, e.g., multiplicative noise, can be found in [33].

It is also interesting to note that equation (2.20) can be cast in the level set formulation that will be discussed in Chapter 3. For the above model the speed function that will be introduced in equation (3.4) is given by

$$F = \frac{\kappa}{|\nabla u|}.$$

This means that all iso-intensity contours in the image move with a speed equal to the curvature of the level sets scaled by an inverse magnitude of the gradient.

2.4.3 A Sigmoid Function

The last function $\phi(s)$ presented here was proposed by Green in [16]. This very interesting choice leads to a nonlinearity in the divergence term containing the sigmoid function $\tanh(s)$. We will return to this choice again in Chapter 4 and 5. The PDE associated to the minimization problem using $\phi(s) = \log(\cosh(s))$ is given by

$$\begin{aligned} \hat{u}_t &= \operatorname{div}\left(\frac{\tanh(|\nabla\hat{u}|/K)}{|\nabla\hat{u}|}\nabla\hat{u}\right) + (u_0 - \hat{u}) \\ \hat{u}(x,y,0) &= u_0(x,y). \end{aligned} \tag{2.22}$$

Figure 2-11 shows the test image with $\mathrm{SNR}(u/u_0) = 3.4$ and the steady-state solution of equation (2.22) with $K = 0.02$. The result has $\mathrm{SNR}(u/\hat{u}) = 21.0$. Compared to the enhancement schemes discussed here the performance is equivalent to the total variation minimization and, as we will see in the next chapter, to the curvature flow of Alvarez *et al.*

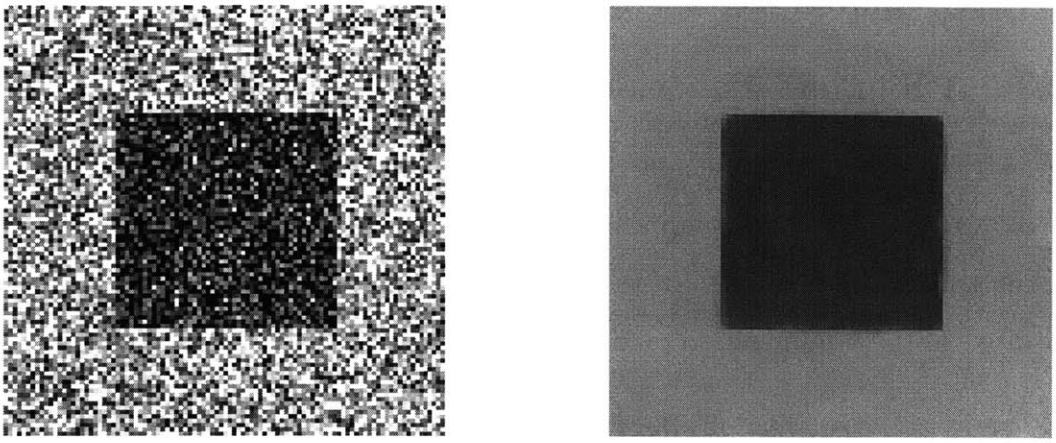

Figure 2-10: Original image on the left and result obtained from the total variation minimization.

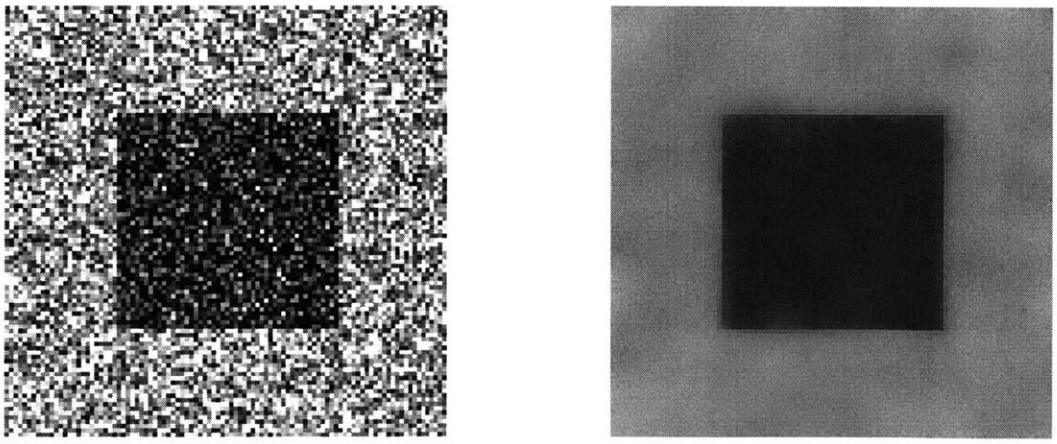

Figure 2-11: Test image on the left and filtered image obtained from equation (2.22).

Chapter 3

Curvature Flows and Shock Filters

3.1 Introduction

This chapter presents two more approaches to image enhancement using PDE's.

The first part of this chapter introduces a method that makes use of curvature flows to enhance images. In this context the image is interpreted as a collection of curves, strictly speaking the *iso-intensity contours* of the image, which can be evolved. Before stating the image processing algorithm however, a thorough discussion of curve evolution and level sets is necessary for understanding the ideas.

In the second part we present a method known as shock filters. Since it borrows ideas from fluid dynamics we will first discuss the ability of certain partial differential equations to have discontinuous solutions and to build shocks. Afterwards the application to image enhancement is shown.

3.2 Curve Evolution and Level Sets

In this section we will consider the problem of evolving a curve normal to itself with a certain speed. Let us assume that $C_0(s), s \in [0, L(C)]$ is a closed parameterized planar curve, which moves in the direction of its inward Euclidean normal vector $\vec{\mathcal{N}}$ with a speed given by the speed function F. Using this notation the evolution of the

curves $C(s,t)$ can be written as

$$\begin{aligned} \frac{\partial C(s,t)}{\partial t} &= F \cdot \vec{\mathcal{N}} \\ C(s,t=0) &= C_0(s). \end{aligned} \tag{3.1}$$

A standard numerical method to solve the curve evolution equation is the marker particle method. In this method N markers are placed on the initial curve in regular intervals. The equation of motion associated to the curve evolution is then solved separately for each marker $(x_i, y_i) = C(s_i)$, $i = 1, \ldots, \mathrm{N}$. The main drawback of this approach is that the evolving curve cannot deal with topological changes of the moving front, e.g. splitting, merging and crossing over itself. This can be avoided by a method introduced by Osher and Sethian [30, 38] called level set methods.

In this approach the initial curve is represented by the zero level set $(\phi(x,y) = 0)$ of an initial surface $\phi_0(x,y)$. The surface is defined at each point (x,y) by $\phi_0(x,y) = \pm d(x,y)$, where $d(x,y)$ is the distance from (x,y) to the initial contour $C_0(s)$. The minus sign is used if (x,y) is inside $C_0(s)$ and the plus sign if the point is outside. At time T the evolved curve $C(s,T)$ is described by the points (x,y) which satisfy the equation $\phi(x,y,T) = 0$, also called the zero level set of ϕ. We can now derive the evolution equation for the surface $\phi(x,y,t)$.

The speed of each point (x,y) on the curve in the direction $\vec{\mathcal{N}}$ normal to the zero level set is given by the speed function F. Thus

$$\frac{\partial C(s,t)}{\partial t} \cdot \vec{\mathcal{N}} = F,$$

where $\frac{\partial C(s,t)}{\partial t} = (\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t})$. The inward normal vector is defined by

$$\vec{\mathcal{N}} = -\frac{\nabla \phi}{|\nabla \phi|}, \tag{3.2}$$

given the initial ϕ. Differentiating ϕ with respect to t by the chain rule gives

$$\phi_t + \frac{\partial C(s,t)}{\partial t} \cdot \nabla \phi = 0. \tag{3.3}$$

Combining equations (3.1)-(3.3) we can write ϕ as the solution of the partial differential equation

$$\phi_t - F\,|\nabla \phi| = 0, \tag{3.4}$$

$$\phi(x,y,0) = \pm d(x,y). \tag{3.5}$$

At any time, the moving front $C(s,t)$ is just the zero-level set of ϕ, that is

$$C(s,t) \;=\; [\ (x,y) \mid \phi(x,y,t) = 0\ ]. \tag{3.6}$$

Level set methods thus enable us to cast the evolution of a curve defined by equation (3.1) into the initial value partial differential equation (3.4) and (3.5). At first glance it seems contradictory to introduce a new dimension in order to reduce the complexity of the problem. But in fact it is easier to find accurate numerical schemes for equation (3.4) than using marker particle methods. Furthermore, finite difference approximations can be used which exploit techniques borrowed from the numerical solution of hyperbolic conservation laws. Finally, level set methods can handle topological changes of the moving front as mentioned earlier.

The image enhancement schemes to be introduced later can be put in the form of equation (3.4). With the preceding discussion it will therefore be easier to understand how these models work. But first let us have a look at two specific speed functions.

Speed function $F = \pm 1$

The first speed function is simply given by $F = \pm 1$. In this case the equation defines the operators "erosion" and "dilation" of classical mathematical morphology. These operators can also be used for image processing but this topic is beyond the scope of this thesis. A more detailed discussion may be found in [41, 42].

Speed function $F = \kappa$

The second speed function is more interesting for us and is given by $F = \kappa$, where κ is the curvature of the moving front. With our previous definition of ϕ the curvature is obtained from the divergence of the unit normal vector to the front

$$\kappa = \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right), \tag{3.7}$$

and is positive on all convex level curves. Since F controls the speed in the direction of the inward normal, a curve evolving with speed $F = \kappa$ shortens and shrinks. This

flow is known as the curve shortening or mean curvature flow[1] and corresponds to a curve collapsing under its curvature. Gage and Hamilton [13, 14] showed that a convex curve moving under this speed function must evolve to a round point without developing self-intersections or singularities. Grayson [15] generalized this result and proved that any nonconvex curve will evolve into a convex curve without crossing over itself or forming edges in the front, that is, without developing self-intersections or singularities. Combining these two results means that any arbitrary simple closed curve moving under its curvature collapses in finite time nicely to a round point and then disappears.

We should note here that curvature motion has important smoothing properties. Flow under curvature smooths out oscillations in the front and removes spikes of noise since they correspond to high curvature objects. Thus small oscillations corresponding to high curvature will disappear before the overall shape is altered. The deformation of a curve by its curvature given by equation (3.1) with $F = \kappa$ is therefore also known as the *geometric heat equation.*

So far we have only considered the evolution of a curve if we construct ϕ_0 as the signed distance function to the initial curve. In the context of image processing we would have to construct an initial surface for every iso-intensity contour in the image which would not be applicable in practice. Fortunately, Kimia *et al* [20] showed that the evolution of $C_0(s)$ into $C(s,t)$ is independent of the choice of the initial surface ϕ_0. Thus for our intensity image $u(x,y)$ it is mathematically valid to directly obtain the surface ϕ from the gray-level information and to evolve the whole image at once. This is equivalent to evolving each iso-intensity curve separately by curvature and then combining them again.

Before moving on to an image enhancement scheme based on curve evolution it should be noted that level set methods can also be used for other image processing tasks, such as *geodesic active contours* [6] or *snakes* [19].

[1]The level sets of the solution move in the normal direction with a speed proportional to their mean curvature [9], thus the term "mean curvature motion" effect.

3.3 Image Selective Smoothing by Nonlinear Diffusion II

Curvature flows in image processing were first introduced in Alvarez *et al* [2]. They proposed to obtain an estimate $\hat{u}(x,y)$ of $u_0(x,y)$ by solving

$$\begin{aligned} \hat{u}_t &= g(|\nabla G_\sigma * \hat{u}|)\ |\nabla \hat{u}|\ \mathrm{div}\left(\frac{\nabla \hat{u}}{|\nabla \hat{u}|}\right) \\ \hat{u}(x,y,0) &= u_0(x,y), \end{aligned} \tag{3.8}$$

where $g(s)$ is defined as in equation (2.4) or equation (2.5). Using equation (1.11) we can rewrite the above equation in the form of equation (1.12) and obtain

$$\hat{u}_t = g(|\nabla G_\sigma * \hat{u}|)\ \hat{u}_{\xi\xi}. \tag{3.9}$$

Recall from Section 1.2 that $\hat{u}_{\xi\xi}$ is a diffusion term which diffuses $\hat{u}(x,y)$ only in the direction ξ orthogonal to its gradient and not at all in the direction of $\nabla\hat{u}$. Thus, the aim of this model is to diffuse $\hat{u}(x,y)$ on both sides of an edge with minimal smoothing of the edge itself. The term $g(|\nabla G_\sigma * \hat{u}|)$ controls the speed of the diffusion[2] . If $|\nabla G_\sigma * \hat{u}|$ is small in a neighbourhood of a point x this point is considered to lie in the interior of a smooth region of the image and the diffusion is therefore strong. If $|\nabla G_\sigma * \hat{u}|$ has a large value, the point is considered as an edge point and the diffusion is lowered, since $g(s)$ is small for large s.

With the previous introduction of level set methods there is also another interpretation. The main point is to consider the image as a collection of iso-intensity contours. Since we are free in the choice of our initial surface we can simply take the gray-level image $u_0(x,y)$ as the initial surface. Thus we can evolve the whole image $u_0(x,y)$ by curvature flow and know that the same holds true for every iso-intensity contour. Using equation (3.7) we can write equation (3.8) in the form

$$\hat{u}_t = g(|\nabla G_\sigma * \hat{u}|)\ \kappa\ |\nabla \hat{u}|. \tag{3.10}$$

[2]Note that $(G_\sigma * \hat{u})$ is a filtered version of $\hat{u}$ and is therefore more robust in the presence of high noise.

Comparing this equation with equation (3.4) we see that the speed function for the curve evolution is given by $F = g(|\nabla G_\sigma * \hat{u}|)\ \kappa$ here. That means that each iso-intensity contour of the image moves with speed $g(|\nabla G_\sigma * \hat{u}|)\ \kappa$.

Let us have a closer look at the properties of equation (3.10). The most important advantage of curvature flows is that it does not alter the contrast between different shapes. There is no smoothing or diffusion taking place across the level lines of the image. Thus boundaries remain sharp and do not blur, smoothing takes place inside regions but not across. A disadvantage associated with equation (3.10) is that all information is eventually removed from the image through continued application of the scheme. This is due to the nature of curvature flows described in the last section: each curve is first smoothed by curvature deformation, then it shrinks to zero and disappears. Since the speed is never equal to zero the image will finally disappear, making it necessary to choose a stopping time.

We can now have a look at the filtered image obtained using the method of Alvarez *et al.* Figure 3-1 shows the input image on the left with $\mathrm{SNR}(u/u_0) = 3.4$ and the resulting image on the right with $\mathrm{SNR}(u/\hat{u}) = 24.2$. The stopping function g is given by equation (2.5) with $K = 0.6$. The Gaussian function used has a variance $\sigma = 2$ and is truncated by including 12 points on each side of the point at which g is calculated. It seems that the edges of the square are blurred, an effect which comes from the numerical approximation of the scheme.

Alvarez *et al* proposed a modification of this model in [2]. The reason is obvious if we recall equation (3.9) and its interpretation: the diffusion in the image takes place only perpendicular to the gradient. This property is desirable along edges but it is not necessary to diffuse anisotropically at points where the gradient is low. Inside a region of uniform brightness the scheme should diffuse in both directions. Therefore a better model is

$$\hat{u}_t = g(|\nabla G_\sigma * \hat{u}|)\ \left[(1 - k(|\nabla \hat{u}|))\ \Delta \hat{u}\ +\ k(|\nabla \hat{u}|)|\nabla \hat{u}|\ \mathrm{div}\left(\frac{\nabla \hat{u}}{|\nabla \hat{u}|}\right)\right]\ , \qquad (3.11)$$

where $k(s)$ is a smooth nondecreasing function such that $k(s) = 0$ if $s \leq e$, $k(s) = 1$ if $s \geq 2e$. The parameter e introduced here is not a new parameter. The function $k(s)$

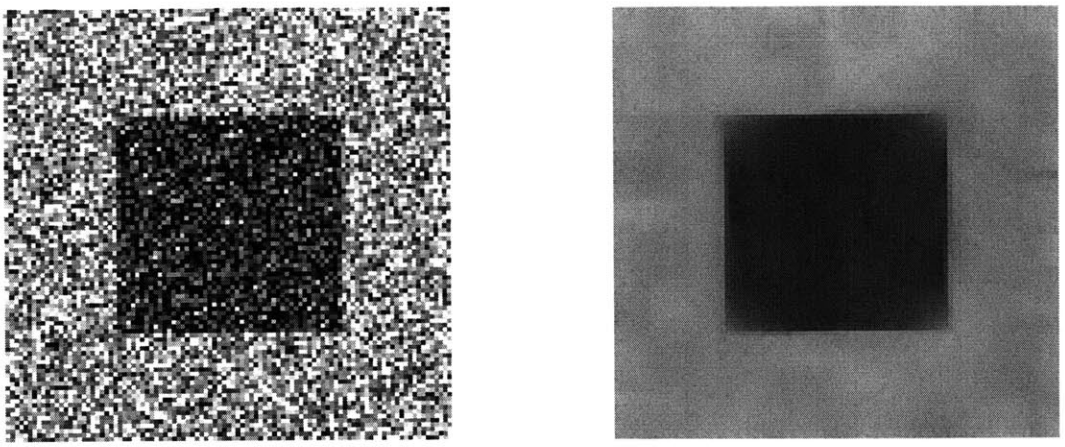

Figure 3-1: Original image on the left and solution of the curvature flow of Alvarez *et al.*

is used to control the direction of diffusion and is similar to the stopping function $g(s)$. Thus e depends on the parameters K and α of $g(s)$. Rewriting the modified equation we obtain

$$\hat{u}_t = g(|\nabla G_\sigma * \hat{u}|)\hat{u}_{\xi\xi} + g(|G_\sigma * \nabla\hat{u}|)(1 - k(|\nabla\hat{u}|))\hat{u}_{\eta\eta}. \tag{3.12}$$

For $|\nabla\hat{u}| \geq 2e$ the second term is zero and the equation reduces to equation (3.9): diffusion only takes place perpendicular to the gradient. Whereas for $|\nabla\hat{u}| \leq e$ the amount of smoothing is the same for both directions. We obtain a scheme that guarantees anisotropic smoothing along edges and isotropic smoothing inside regions of uniform brightness. The input image ($\text{SNR}(u/u_0) = 3.4$) and the result ($\text{SNR}(u/\hat{u}) = 23.8$) obtained by equation (3.11) are shown in Figure 3-2. The stopping function g is given by equation (2.5) with $K = 0.6$ and the parameter $e = 0.03$. The Gaussian function used has a variance $\sigma = 2$ and is truncated in the same way as above. Comparing it with Figure 3-1 we can see that the brightness inside and outside the square is more uniform now.

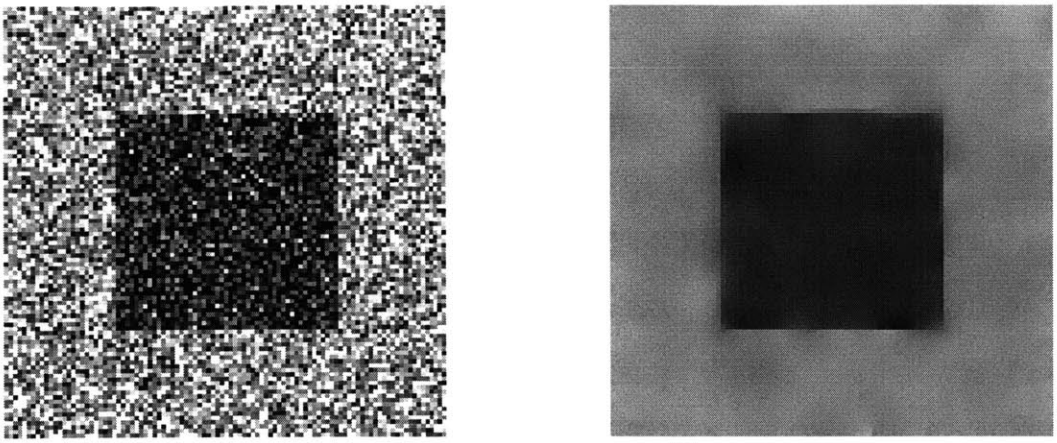

Figure 3-2: Original image on the left and solution of the modified equation of Alvarez *et al.*

3.4 The Burger's Equation

The idea of applying the theory of shock calculations to image processing was introduced by Rudin [32] in which the numerical analysis of singularities and shock filters are studied. Shock waves are well known in fluid dynamics and are closely related to the theory of hyperbolic conservation laws. An extensive discussion of the theory of shocks is beyond the scope of this thesis and we consider only their application to image processing. For further information the reader is referred to [22, 23].

The reason we can use shock calculations in image processing is based on the way images are described. By assigning a brightness value to each pixel we can interpret the image intensity function $u(x, y)$ as a surface over the $x - y$ plane. It was already mentioned in Chapter 1 that an edge in the image is nothing but a jump or discontinuity in the brightness distribution $u(x, y)$. Consider again Figure 2-6 which showed a one-dimensional step edge and its blurred version. Our goal is now to restore the original step edge from the blurred one. This is equivalent to the development of a shock in the brightness function $u(x)$. We can try to achieve this goal by making use of equations similar to those known from shock calculations.

One of the simplest models that demonstrates singular behaviour in its solution

is the inviscid Burger's equation given by

$$u_t + \frac{1}{2}(u^2)_x \;=\; u_t + u \cdot u_x \;=\; 0, \tag{3.13}$$

where $u(x, t = 0) = u_0(x)$. If the initial condition $u_0(x)$ is not monotonically non-decreasing in the space variable x this equation may lead to discontinuous solutions in finite time. More precisely the solution is determined by the characteristics which are defined by $\frac{dx}{dt} = u$. These are curves in the $x - t$ space along which u is constant. The development of a shock then corresponds to the crossing of characteristics. Figure 3-3 shows the development of a right facing profile over time on the left and its characteristics on the right. As we can see from the characteristics the shock starts to form at $t = 1$ sec. and then keeps traveling to the left. The shock speed is controlled by the Rankine-Hugoniot jump condition. Note that if a discontinuity or shock forms the differential equation is no longer valid. In this case the solution is given by the so called weak solution. Weak solutions are essentially solutions that satisfy the differential equation where the solution is smooth, and the jump condition at discontinuities.

Figure 3-4 shows the development of the same profile as in Figure 3-3 but now facing to the left. The characteristics plotted on the right do not cross and therefore no shock develops. In contrast to Figure 3-3 the profile tends to be smeared out.

We can also include viscous effects in the above equation by adding a diffusion term. This leads to the viscous Burger's equation

$$u_t + u \cdot u_x = \varepsilon u_{xx}. \tag{3.14}$$

When the viscous term is included, the equation becomes parabolic and does not admit discontinuous solutions. The second derivative u_{xx} acts like a smoothing term and stops the development of shocks. We can try to achieve a balance by controlling ε: decaying u and at the same time steepening right facing profiles. This could theoretically lead us to edge enhancement and simultaneous smoothing of noise. However, from the above discussion it should be clear that there are problems inherent to equation (3.14) that make it unsuitable for applications to image processing. Let us review these problems and show how to overcome them:

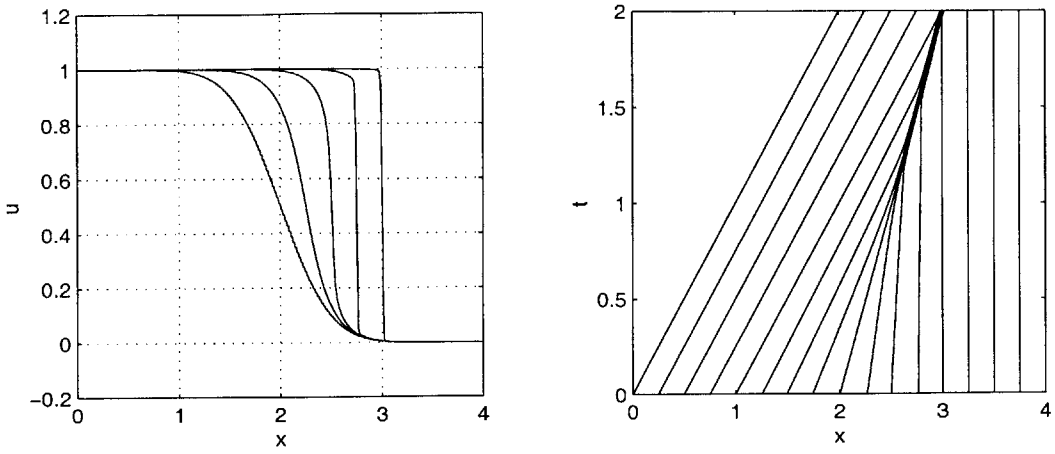

Figure 3-3: Solution of the inviscid Burger's Equation for the right facing profile at times $t = 0, 0.5, 1.0, 1.5$, and 2 sec. on the left and its characteristics on the right.

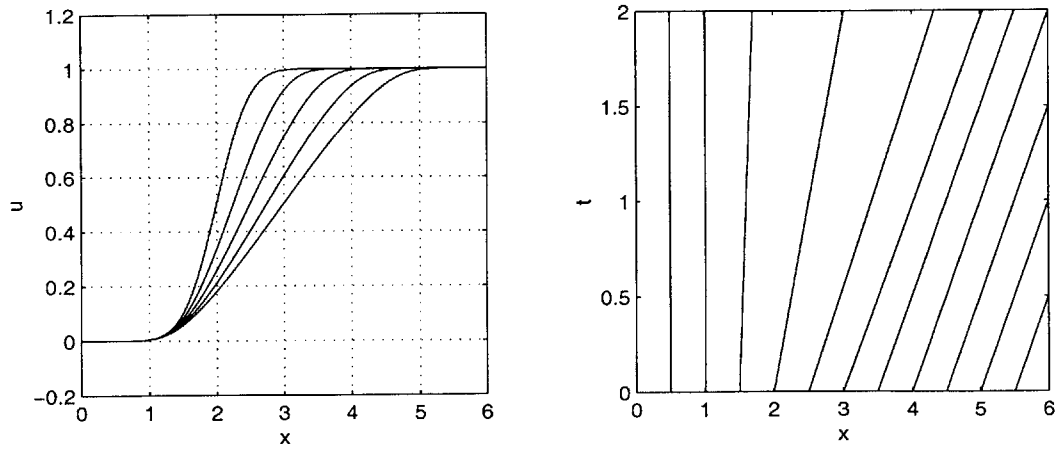

Figure 3-4: Solution of the inviscid Burger's Equation for the left facing profile at times $t = 0, 0.5, 1.0, 1.5$, and 2 sec. on the left and its characteristics on the right.

- Equation 3.14 is not symmetric in the x-space. Thus, all left facing profiles will be smoothed while right facing profiles will be steepened as we saw in the preceding plots. However, by replacing u_x by $|u_x|$ the equation can be modified and put in a space symmetric form.

- The solution is not stationary because different parts of the initial signal travel with different velocities. To prevent this problem the "speed" has to be controlled by further transforming the equation into

$$u_t + f(u_x) \cdot u \cdot |u_x| = \varepsilon u_{xx}, \tag{3.15}$$

  where the coefficient $f(u_x)$ is set to control the diffusion speed.

3.5 Shock Filters

Osher and Rudin [29] introduced a shock filter based on the theory of the last section. First we discuss the filter and its performance in 1-D. The extension to the 2-D filter and its application to images will be straightforward.

The main point when applying shock filters is that we assume the input image to be blurred but *without* noise. Since the filter sharpens edges any noise in the image will also be enhanced by this procedure and the result would be useless. However, we can still apply shock filters to noisy images if we blur these images first. The low-pass filtering can be done by convolving the noisy image with a Gaussian function, e.g., in Section 3.5.2 we will blur our standard test image first and then take the result as the input for the shock filter. Although this way of approaching the problem works for our test image, we will see the use of this procedure is limited. If the noise is very high the diffusion has to be strong to reduce all oscillations. It can therefore happen that too much information is already lost because of the blurring, e.g., two objects or image features that are close to each other might merge.

3.5.1 One-dimensional Shock Filter

The one-dimensional version of the shock filter is given by

$$\begin{aligned} u_t &= -|u_x|\, F(u_{xx}) \\ u(x,0) &= u_0(x). \end{aligned} \tag{3.16}$$

Comparing this filter to equation (3.15) we can see the relationship to Section 3.4. Since we assume the input to be already blurred we do not need to be concerned about smoothing. We can therefore leave out the viscous term and set $\varepsilon = 0$. Thus, the only difference is to replace $(f(u_x) \cdot u)$ by $F(u_{xx})$. F is set to change sign across any essential singular feature and is used as the edge detector switch, e.g., $F = \text{sign}(u_{xx})$[3].

In Figure 3-5 we apply equation (3.16) to a sine wave. The plot shows the wave at times $t = 0, 0.5, 1.0$ sec. and the steady state solution at 2.0 sec. We can see that shocks are formed and that the restored signal will be piecewise smooth.

3.5.2 Two-dimensional Shock Filter

The two-dimensional shock filter used for image enhancement is given by

$$\begin{aligned} u_t &= -\sqrt{u_x^2 + u_y^2}\, F(\mathcal{L}(u)) \\ u(x,y,0) &= u_0(x,y). \end{aligned} \tag{3.17}$$

The edge detector switch is defined by $F = \text{sign}(\mathcal{L}(u))$. The sharpening of edges will now occur at places where $\mathcal{L}(u) = 0$. Compared to the one-dimensional version there is more freedom of choice as to which edge detector $\mathcal{L}(u)$ to use in the 2-D case. The simplest choice for $\mathcal{L}(u)$ is the Laplacian

$$\mathcal{L}(u) = u_{xx} + u_{yy}. \tag{3.18}$$

Another choice for $\mathcal{L}(u)$ follows the idea that the edges are located at the zero-crossings of the second directional derivative of $u(x,y)$ in the direction of the gradient.

[3]Recall from Chapter 2 that edges are located at the zero-crossings of the (one-dimensional) Laplacian in the classical theory.

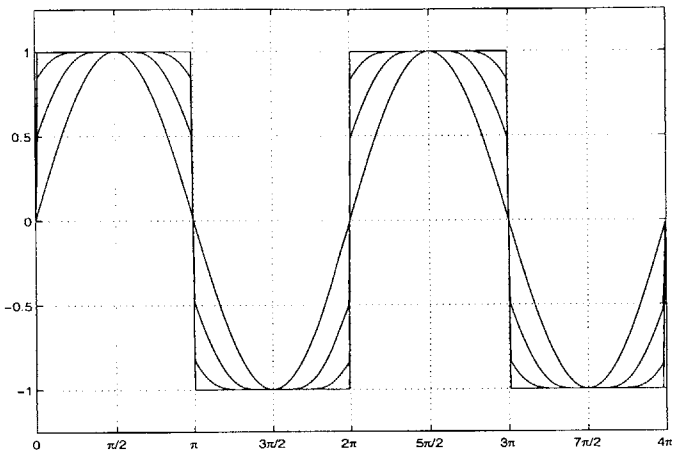

Figure 3-5: One-dimensional shock filter applied to sine wave. The result is shown at times $t = 0, 0.5, 1.0$ and 2.0 sec.

With equation (1.9) the condition that $u_{\eta\eta} = 0$ is equivalent to setting

$$\mathcal{L}(u) = u_{xx}u_x^2 + 2u_{xy}u_xu_y + u_{yy}u_y^2. \tag{3.19}$$

Figure 3-6 shows the noisy test image on the left. In the first step the noisy image is taken as the input to a standard approximation to the diffusion equation with diffusion constant $c = 2$. The number of iterations is 200 and the time-step $\Delta t = 0.05$. The blurred output is shown in the middle. In the second step the shock filter is applied to the blurred image. The time-step $\Delta t = 0.05$, the number of iterations is 250, and $\mathcal{L}(u)$ is taken from equation (3.19). The restored image is plotted on the right. Similar to the 1-D case the restored image is piecewise smooth. Furthermore we should note that the solution reached a steady-state. Choosing a higher number of iterations would produce no further change.

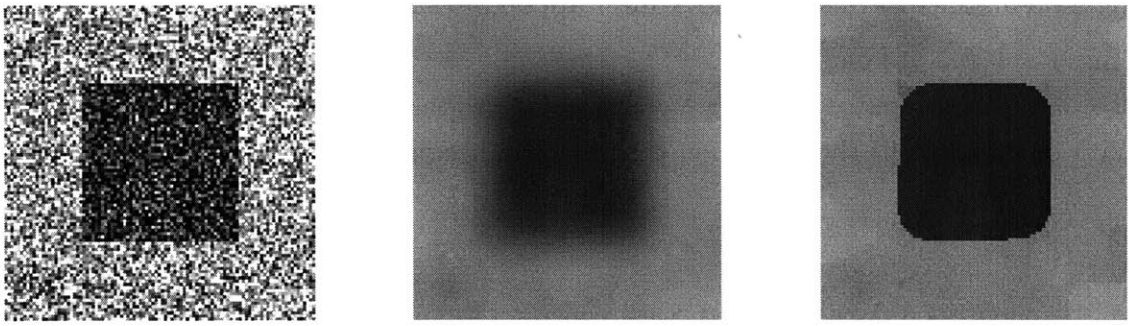

Figure 3-6: From left to right: Noisy input image, blurred version produced by the diffusion equation, and the steady state solution of the shock filter with blurred image taken as input.

Chapter 4

Contraction Analysis applied to Image Processing

4.1 Introduction

This chapter applies recent results from the stability analysis of nonlinear reaction-diffusion equations to image processing. We will first state a result obtained using contraction theory that enables us to quantify exponential convergence rates and to guarantee the global stability of the system.

Afterwards, we will consider the enhancement of one-dimensional noisy signals before extending these ideas to images. The last section shows the discretization used to solve the model equation in one and two spatial dimensions.

4.2 Nonlinear Reaction Diffusion Equations

We will first state a result from the stability analysis of a certain class of nonlinear reaction-diffusion equations which is obtained by extending a recently developed analysis and control system tool called *contraction analysis* to partial differential equations [25]. To quantify the stability properties of the system the right hand side of the PDE is discretized using finite differences. The continuous state-space quantities are then computed as the limits of the discretized versions, as the discretization step

tends to zero. The main advantage of the method is that it guarantees global stability and exponential convergence if the system under consideration is contracting. The result [25]is given below.

Consider the nonlinear reaction-diffusion equation

$$\frac{\partial \phi}{\partial t} = \nabla \mathbf{h}(\nabla \phi, t) + f(\phi, \mathbf{x}, t) \tag{4.1}$$

with continuous state ϕ, position $\mathbf{x}$, time t and where $\partial \mathbf{h}/\partial \nabla \phi \geq 0$. Here, bold characters, as e.g. $\mathbf{h}$, denote vector quantities[1]. We denote by $\max(l_i)$ the diameter (maximum length) of the continuum along the i^{th} axis, and by $\min(\lambda_{\frac{\partial \mathbf{h}}{\partial \nabla \phi}})$ a lower bound on the smallest eigenvalue of the symmetric part of $\frac{\partial \mathbf{h}}{\partial \nabla \phi}$ on the continuum. Assume that

$$\lambda_{\nabla \mathbf{h}} + \max\left(\frac{\partial f}{\partial \phi}\right) \tag{4.2}$$

is uniformly negative, where

$$\lambda_{\nabla \mathbf{h}} = -\min(\lambda_{\frac{\partial \mathbf{h}}{\partial \nabla \phi}}) \sum_{i=1}^{m} \frac{\pi^2}{\max(l_i)^2} \tag{4.3}$$

for a Dirichlet condition ($\phi(t)$ specified on the boundary), and

$$\lambda_{\nabla \mathbf{h}} = 0 \tag{4.4}$$

for a Neumann condition ($\nabla \phi(t)$ specified on the boundary). Then, all system trajectories converge exponentially to a single field $\phi_d(\mathbf{x}, t)$, with minimal convergence rate $|\lambda_{\nabla \mathbf{h}} + \max(\frac{\partial f}{\partial \phi})|$.

In the autonomous case ($f = f(\phi, \mathbf{x})$, and with constant boundary conditions) the system converges exponentially to a steady-state $\phi_d(\mathbf{x})$, which is the unique solution of the generalized Poisson equation

$$0 = \nabla \mathbf{h}(\nabla \phi_d) + f(\phi_d, \mathbf{x}).$$

Let us now show how these concepts and results can be applied in the context of image enhancement. We will first discuss the enhancement of one-dimensional signals before moving on to two-dimensonal signal and image enhancement [26].

[1]Note that $\nabla \mathbf{h} = \operatorname{div} \mathbf{h}$ since $\mathbf{h}$ is a vector quantity.

4.3 One-dimensional Signal Enhancement

We consider a general one-dimensional reaction-diffusion equation of the form

$$\hat{u}_t = \text{div } h(\nabla\hat{u}, t) + f(\hat{u}, x, t), \qquad \text{div } h = \frac{\partial}{\partial x}(h) \tag{4.5}$$

with Neumann boundary conditions $\nabla\hat{u} = 0$. In the 1-D case $\nabla\hat{u}$ reduces to $\hat{u}_x$ and therefore $h(\nabla\hat{u}, t)$ is just a scalar function. Let us review the conditions we have to satisfy for this equation to be contracting. According to the last section $\min(\lambda_{\frac{\partial h}{\partial \nabla \hat{u}}})$ has to be nonnegative and f strictly decreasing. Since h is just a scalar function the first condition is equivalent to requiring that $\frac{\partial h}{\partial \nabla \hat{u}} \geq 0$. Thus, choosing any arbitrary nondecreasing function h and strictly decreasing function f will guarantee the global stability of equation (4.5). On the other hand it also depends on the shape of h and f if we can achieve noise removal while keeping edges intact. Let us first use an intuitive approach to find choices of h and f that meet both requirements simultaneously.

An intuitive approach

Taking a look at the reaction term $f(\hat{u}, x, t)$, we want our choice for f to ensure that the solution of our enhancement scheme converges to the desired restored signal. Thus, we will take f to be a simple feedback of the form $f = (u_0 - \hat{u})$, where u_0 is the original signal. This choice penalizes the deviation from the original signal and guarantees that the solution reaches a non-trivial steady-state.

The determination of $h(\nabla\hat{u}, t)$ is more difficult but we can follow the ideas presented in Chapter 2 and set the function h inside the divergence operator to control the diffusion. First of all assume a separable pattern of image and noise characteristics. This implies that oscillations in the gradient $\nabla\hat{u}$ that are smaller than a certain threshold θ correspond to noise while features where $\nabla\hat{u}$ is greater than θ correspond to edges. Thus we want to find an h that stops the diffusion and gives div $h(\nabla\hat{u}) = 0$ for $\nabla\hat{u} > \theta$, while removing the oscillations and giving div $h(\nabla\hat{u}) \neq 0$ for $\nabla\hat{u} < \theta$. Formulating this condition for h we want to have $h(\nabla\hat{u}) =$ constant for $\nabla\hat{u} > \theta$, thus satisfying the condition div $h(\nabla\hat{u}) = 0$ for this range of $\nabla\hat{u}$. Below the threshold we want h to vary and ensure that div $h(\nabla\hat{u}) \neq 0$.

A smooth saturation function would therefore be a natural choice of $h(\nabla\hat{u})$, i.e., one could take

$$h(\nabla\hat{u}) = \sin\left(\frac{\pi}{2}\,\mathrm{sat}\left(\frac{\nabla\hat{u}}{K}\right)\right), \tag{4.6}$$

where "sat" is the saturation[2]. The constant K is used to set the threshold θ above which h = constant. In the above equation θ is related to K simply by $\theta = K$. The shape for this choice of h is shown on the left of Figure 4-1 for $K = 3$. Another possibility for h would be

$$h(\nabla\hat{u}) = \tanh\left(\frac{\nabla\hat{u}}{K}\right). \tag{4.7}$$

In this case there is no exact value for which h = constant. Indeed, the derivative of h will never be exactly equal to zero. However, a threshold for which this is approximately true is given by $\theta \approx 3K$ as we can see in Figure 4-1 on the right for $K = 1$.

Let us now check how those choices of h and f perform on a noisy test signal. Figure 4-2 shows a noisy signal and its estimate produced by equation (4.5). We used $h(\nabla\hat{u})$ from equation (4.6) with $K = 0.05$. The plot shows the steady-state solution of our model.

We will show that there is actually a justification for the above choices that is based on a variational formulation.

A variational approach

Let us first point out the resemblance between the reaction-diffusion equation (4.5) and the partial differential equation (2.10) obtained from the variational formulation

[2]The saturation is defined as

$$\mathrm{sat}\left(\frac{x}{K}\right) = \begin{cases} 1 & \text{for } x > K, \\ \frac{x}{K} & \text{for } -K < x < K, \\ -1 & \text{for } x < -K. \end{cases}$$

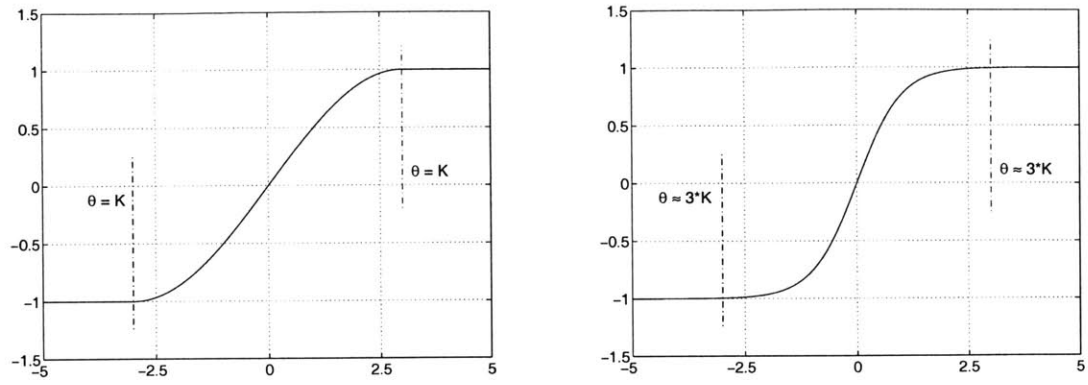

Figure 4-1: Possible choices for the function $h(\nabla\hat{u})$ as the argument of the divergence operator.

in Chapter 2. If we state a similar minimization problem like the one in equation (2.8) for the one-dimensional case we get

$$\inf_{\hat{u}} \left\{ \int_x \left(\frac{1}{2} (\hat{u} - u_0)^2 + \phi(\nabla\hat{u}) \right) dx \right\}, \tag{4.8}$$

and solving the Euler-Lagrange equation of this problem using a steepest descent method leads to the partial differential equation

$$\begin{aligned} \hat{u}_t &= \text{div } \phi'(\nabla\hat{u}) + (u_0 - \hat{u}) \\ \hat{u}(x,0) &= u_0(x). \end{aligned} \tag{4.9}$$

Comparing equation (4.5) and equation (4.9) we see that $h(\nabla\hat{u}) = \phi'(\nabla\hat{u})$ and the solution obtained from equation (4.5) is the unique minimizer $\hat{u}$ of equation (4.8) if ϕ is either given by

$$\phi(s) = \log(\cosh(s)) \tag{4.10}$$

or

$$\phi(s) = \begin{cases} s - \frac{\pi-2}{\pi} & \text{if } s \geq 1 \\ \frac{2}{\pi}\left(1 - \cos\left(\frac{\pi}{2}s\right)\right) & \text{if } s < 1, \end{cases} \tag{4.11}$$

depending on the choice of $h(s)$. Note that the first choice of ϕ was already mentioned in Chapter 2 and was first proposed by Green [16]. Also note that the condition for

the system (4.5) to be contracting was given by $\frac{\partial h}{\partial \nabla \hat{u}} \geq 0$. This however is equivalent to the condition that ϕ has to be convex, $\phi''(s) \geq 0$; the same requirement that has to hold so that the minimization problem (4.8) has a unique and existing solution [3, 7].

Finally, notice that we can rewrite div $\phi'(\nabla \hat{u})$ as $\phi''(\nabla \hat{u})\ \Delta u$. The diffusion therefore stops for high gradients, which is the one-dimensional equivalent to the conditions (1.13) and (2.13) stated for optimal enhancment in 2-D.

4.4 Two-dimensional Signal and Image Enhancement

The extension of the above results to two-dimensional signal and image enhancement is straightforward. Let us first review the conditions that must be satisfied for the reaction diffusion equation to be contracting in 2-D. The equation is given by

$$u_t = \text{div } \mathbf{h}(\nabla u, t) + f(u, \mathbf{x}, t), \tag{4.12}$$

where we only take Neumann boundary conditions $\nabla u = 0$ into account. Note that we use bold characters for $\mathbf{x}$ and $\mathbf{h}$ now since they denote vector quantities.

According to Section 4.2 equation (4.12) is contracting if the function f is strictly decreasing and the field $\mathbf{h}$ has a positive semi-definite Jacobian $\frac{\partial \mathbf{h}(\nabla u)}{\partial \nabla u}$. Choosing a feedback term $(u_0 - \hat{u})$ for f clearly satisfies the first requirement. The choice for $\mathbf{h}$ will be guided by our discussion in Chapter 2 where a typical form of the nonlinear diffusion term was given by

$$\mathbf{h}(\nabla u) = g(s)\nabla u, \quad \text{with } s = |\nabla u|.$$

The Jacobian of $\mathbf{h}$ has to be positive semi-definite for the system to be contracting. Here, the Jacobian is symmetric and can be written as

$$\frac{\partial \mathbf{h}}{\partial \nabla u} = \begin{pmatrix} g(|\nabla u|) & 0 \\ 0 & g(|\nabla u|) \end{pmatrix} + \frac{\partial g(s)}{\partial s} \frac{1}{|\nabla u|} \begin{pmatrix} u_x^2 & u_x u_y \\ u_x u_y & u_y^2 \end{pmatrix}$$

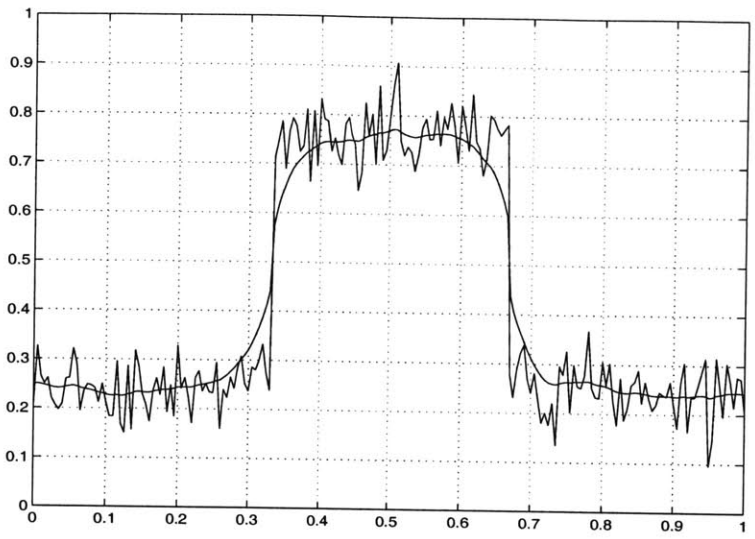

Figure 4-2: Noisy signal and its estimate produced by equation (4.5).

or

$$\begin{aligned}\frac{\partial \mathbf{h}}{\partial(\nabla u)} &= g(s)\,\mathbf{I} + \frac{\partial g(s)}{\partial s}\,\frac{\nabla u \nabla u^T}{|\nabla u|} \\ &= g(s)\,\mathbf{I} + s\,\frac{\partial g(s)}{\partial s}\,\frac{\nabla u}{|\nabla u|}\,\frac{\nabla u^T}{|\nabla u|}.\end{aligned}$$

Since the largest eigenvalue of the dyadic product of two unit vectors is 1, the condition that must be satisfied is

$$g(s) + s\,\frac{\partial g(s)}{\partial s} = \frac{\partial}{\partial s}(s\,g(s)) \geq 0.$$

Hence, $\lambda_{\nabla \mathbf{h}}$ is negative and the system is globally contracting for $g \geq 0$ and $\frac{\partial(sg(s))}{\partial s} \geq 0$. By considering the saturated Laplace operator $\mathrm{div}(g(|\nabla u|)\,\nabla u)$ we reformulated the condition $\mathbf{h}$ has to satisfy as a condition for the scalar function g.

We can now extend the results from 1-D to 2-D in the following way: we solve the two-dimensional variational problem (2.8) using the penalty function $\phi(|\nabla \hat{u}|)$ of equation (4.11)[3]. Similar to Chapter 2 the estimate $\hat{u}$ is given by the steady-state solution of

$$\begin{aligned}\hat{u}_t &= \mathrm{div}(g(|\nabla \hat{u}|)\,\nabla \hat{u}) + (u_0 - \hat{u}) \qquad (4.13)\\ \hat{u}(x,y,0) &= u_0(x,y),\end{aligned}$$

[3]Note that $\phi(s) = \log(\cosh(s))$ has already been discussed in section 2.4.3.

where $g(|\nabla\hat{u}|)$ is given by

$$g(|\nabla\hat{u}|) = \frac{\phi'(|\nabla\hat{u}|)}{|\nabla\hat{u}|} = \frac{\sin\ (\frac{\pi}{2}\ \mathrm{sat}(\frac{|\nabla\hat{u}|}{K}))}{|\nabla\hat{u}|}, \tag{4.14}$$

and the constant K is set to control the diffusion. Note that the system is contracting since $g(s)$ satisfies $g(s) \geq 0$ and $\frac{\partial(sg(s))}{\partial s} \geq 0$. We also note that the condition in terms of ϕ for equation (4.13) to be contracting is given by $\phi'' \geq 0$, thus ϕ has to be convex.

Let us check how the enhancement scheme performs on the test image. Figure 4-3 shows the test image and the steady-state solution of equation (4.13) with g taken from equation (4.14) and $K = 0.05$. The signal to noise ratio of the noisy image is $\mathrm{SNR}(u/u_0) = 3.4$ and that of the result is $\mathrm{SNR}(u/\hat{u}) = 18.0$. Compared to the enhancement schemes discussed earlier the performance is equivalent to the schemes discussed in Sections 2.4.2 and 2.4.3 and to the curvature flow of Alvarez *et al.*

4.4.1 An Alternative Interpretation

We can further analyze the enhancement procedure by transforming it in the form of equation (1.12). Using $\eta - \xi$ coordinates equation (4.13) can be written as

$$\hat{u}_t = g(|\nabla\hat{u}|)\ \hat{u}_{\xi\xi} + G'(|\nabla\hat{u}|)\ \hat{u}_{\eta\eta} + (u_0 - \hat{u}), \tag{4.15}$$

where $G(s) = s \cdot g(s)$. Thus, the diffusion coefficients c_ξ and c_η are given by $c_\xi = g(|\nabla\hat{u}|)$ and $c_\eta = G'(|\nabla\hat{u}|)$, respectively. We can easily check that the conditions (1.13) and (2.13) for optimal enhancement are indeed satisfied for the above choice of $g(s)$. The smoothing takes place in both directions for low gradients whereas for high gradients the diffusion is only in the direction perpendicular to the gradient.

Let us assume now that $|\nabla\hat{u}| > K$ and drop the feedback term for a moment. Since $G'(|\nabla\hat{u}|) = 0$ in this case equation (4.15) reduces to

$$\hat{u}_t = g(|\nabla\hat{u}|)\ \hat{u}_{\xi\xi} = g(|\nabla\hat{u}|)\ \kappa\ |\nabla\hat{u}|. \tag{4.16}$$

Comparing this form to equation (3.4) we notice that the model reduces to a curvature flow for high gradients $|\nabla\hat{u}|$. Recalling the discussion about curvature flows in Chapter 3 we know that all iso-intensity contours of the image move with speed $g(|\nabla\hat{u}|)\ \kappa$.

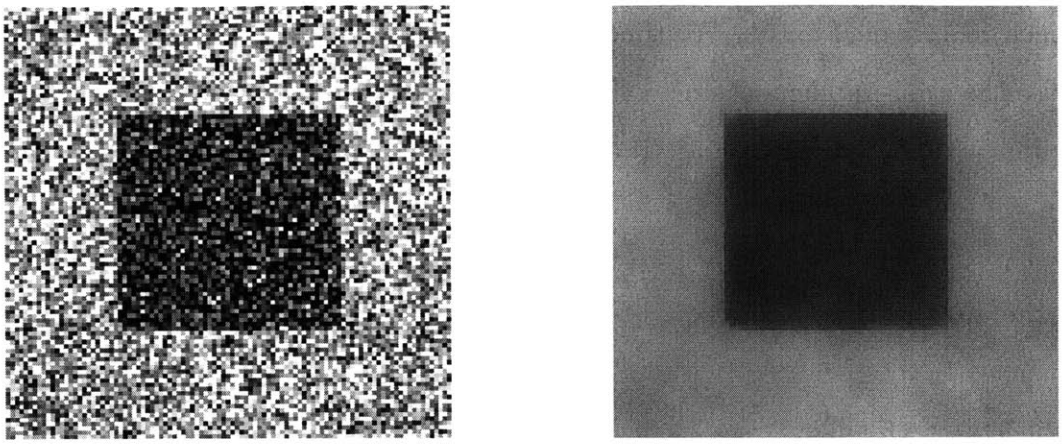

Figure 4-3: Test image on the left and filtered image produced by equation (4.13).

No smoothing or diffusion takes place across the level lines and the contrast between different objects is therefore not altered. Furthermore, boundaries remain sharp and do not blur. This is the reason why the performance of the enhancement scheme is very good even for small K. At the beginning of the enhancement procedure the whole image is smoothed by curvature flow until the strong oscillations are removed. While the model remains a curvature flow across the boundaries all of the time, at one point the diffusion in the second direction is "turned on" within uniform regions, leading to isotropic smoothing and giving a better result. Finally, note that because of the feedback term the equation reaches a non-trivial steady state. The solution converges to the desired restored image.

4.5 Discretization

In this section we outline the discretization scheme used in all the examples above. Equation (4.13) is discretized on a square lattice, with the brightness values $u(x, y, t)$ associated to the vertices, and the conduction coefficients $g(s)$ associated to the arcs. The lattice has a spatial stepsize h and contains N discrete points in each direction, the timestep is denoted Δt. We propose two different schemes for the discretization here: a fully explicit scheme and a semi-implicit one. If we use the explicit scheme

a step size requirement arises in Δt to ensure stability. For the semi-implicit scheme no such requirement exists. For time critical calculations it might be of advantage to choose the semi-implicit scheme.

The image domain Ω is discretized as follows:

$$x_i = ih, \quad y_j = jh, \quad i, j = 1, 2, \ldots, N$$

and the discrete timesteps are

$$t_n = n\ \Delta t, \quad n = 0, 1, \ldots$$

The brightness value $u(x, y, t)$ at each pixel is given by its corresponding value on the grid point, hence

$$u^n_{i,j} = u(x_i, y_j, t_n).$$

The Neumann boundary conditions are implemented using fictitious grid points surrounding the image domain. Denoting those points with indices 0 and $N + 1$ we can write

$$\begin{aligned} u_{0,j} = u_{1,j}, \quad u_{N,j} = u_{N+1,j}, \quad j = 1, \ldots, N \\ u_{i,0} = u_{i,1}, \quad u_{i,N} = u_{i,N+1}, \quad i = 1, \ldots, N. \end{aligned}$$

The continuous derivatives are approximated by finite differences. We will use the following notation for nearest neighborhood differences

$$\begin{aligned} \Delta^x_{\mp} u_{i,j} &= \mp(u_{i\mp1,j} - u_{i,j}) \\ \Delta^y_{\mp} u_{i,j} &= \mp(u_{i,j\mp1} - u_{i,j}). \end{aligned}$$

The conduction coefficients are then given by

$$\begin{aligned} c^n_{N_{i,j}} &= g(|\Delta^x_+ u^n_{i,j}|) \\ c^n_{S_{i,j}} &= g(|\Delta^x_- u^n_{i,j}|) \\ c^n_{E_{i,j}} &= g(|\Delta^y_+ u^n_{i,j}|) \\ c^n_{W_{i,j}} &= g(|\Delta^y_- u^n_{i,j}|) \end{aligned}$$

where $g(s)$ is either given by equation (4.14) or by one of the choices in Table (2.1). The abbreviations N, S, E, and W stand for "North, South, East, and West," respectively. They are defined with respect to the grid point (i, j). The partial derivative with respect to time u_t is discretized by a forward difference scheme

$$u_t = \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t}.$$

Using a 4-point nearest neighborhood discretization of the Laplacian the explicit scheme for the discrete version of equation (4.13) can be written as

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} = \lambda[c_N \cdot \Delta_+^x u - c_S \cdot \Delta_-^x u + c_E \cdot \Delta_+^y u - c_W \cdot \Delta_-^y u]_{i,j}^n + (u_{0_{i,j}} - u_{i,j}^n),$$

where $\lambda = \frac{1}{h^2}$, $i, j = 1, 2, \ldots, N$ and $n = 0, 1, \ldots$. This equation can be put into the form

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} + A_h(u_{i,j}^n)u_{i,j}^n = 0, \quad n \geq 0. \tag{4.17}$$

A step size restriction is imposed for stability

$$\frac{\Delta t}{h^2} \leq c.$$

For the semi-implicit scheme the conduction coefficients are calculated at the old timestep n while new values are used for the nearest neighborhood difference. This gives

$$\begin{aligned}\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} &= \lambda[c_{N_{i,j}}^n \cdot \Delta_+^x u_{i,j}^{n+1} - c_{S_{i,j}}^n \cdot \Delta_-^x u_{i,j}^{n+1} \\ &\quad + c_{E_{i,j}}^n \cdot \Delta_+^y u_{i,j}^{n+1} - c_{W_{i,j}}^n \cdot \Delta_-^y u_{i,j}^{n+1}] + (u_{0_{i,j}} - u_{i,j}^{n+1})\end{aligned}$$

Thus, we can write the semi-implicit scheme as

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} + A_h(u_{i,j}^n)u_{i,j}^{n+1} = 0, \quad n \geq 0, \tag{4.18}$$

where the matrix A_h is tridiagonal by blocks and positive definite. Therefore $(I + \Delta t A_h(u_{i,j}^n))$ is invertible. Equation (4.18) has to hold for each grid point $i, j = 1, 2, \ldots, N$. It gives a linear system of equations that has to be solved at each timestep.

Chapter 5

Image Sequence Enhancement

5.1 Introduction

This chapter is divided into two main parts. In the first part we will show how the results for image enhancement obtained in the previous chapters can be extended to the enhancement of image sequences. We will first discuss the time-invariant noise case and then the time-varying noise case.

In the second part we will propose a new algorithm for image sequence enhancement with time-varying noise. We will start by considering a three-dimensional variational formulation and then propose a fast and recursive solution to this problem.

5.2 Extension of 2-D Image Enhancment to 3-D Image Sequence Enhancement

Before extending the enhancement ideas from static images to time-varying image sequences let us review the model we stated in Chapter 1 to describe image sequences. The first model was defined in equation (1.2) where we assumed that the noise is time-invariant. Since its applicability in practice is very limited we also introduced a more realistic model in equation (1.3) where the noise is time-varying and possibly uncorrelated in time. We will use the first model only at the beginning of this chapter

in Section 5.2.1 and then move on to the second model which is utilized for the remainder of this chapter.

A snapshot (frame number 30) of the synthetic image sequence we will use to test the performance of the algorithms presented is shown in Figure 5-1. The variance of the noise added to the clean image is $\sigma = 0.1$. Only in section 5.2.2 will we also consider cases where the variance of the noise is smaller. The gray-value of the objects in the sequence is 0 and the one of the background is 1. The square on the bottom left moves horizontally to the right while the square on the top right moves vertically downwards into the image. The square on the top left is static and does not move at all.

5.2.1 The constant noise case

One-dimensional signals

Let us start by considering the enhancement of a one-dimensional time-varying signal $u_0(x,t)$ with time-invariant noise ($n \neq n(t)$). Since the enhancement procedure discussed in the last chapter contains a feedback term we can simply apply it to time-varying signals. Recall that the scheme was given by equation (4.5) and that we used a smooth saturation function in the divergence operator. The only difference is that u_0 is now also a function of time, $u_0 = u_0(x,t)$.

This is illustrated in Figure 5-2, which shows intermediate values of the noisy signal and its estimate from zero initial conditions at times $t = 0$, 2.5, and 5.0 sec. It seems that the estimate cannot follow the original signal "fast enough". However, choosing a higher feedback gain, e.g., replacing the feedback term $f = (u_0 - \hat{u})$ by $f = \alpha(u_0 - \hat{u})$, where $\alpha > 1$, also amplifies any noise present in the signal and the result is still noisy.

Thus, instead of modifying the model itself we use a coordinate error feedback. Denoting the estimate of the time-varying signal u_0 by $\hat{u}$ we can write [24]

$$\begin{aligned} \bar{u}_t &= \text{div } h(\nabla\hat{u}) + (u_0 - \hat{u}) \\ \hat{u} &= \bar{u} + u_0. \end{aligned} \tag{5.1}$$

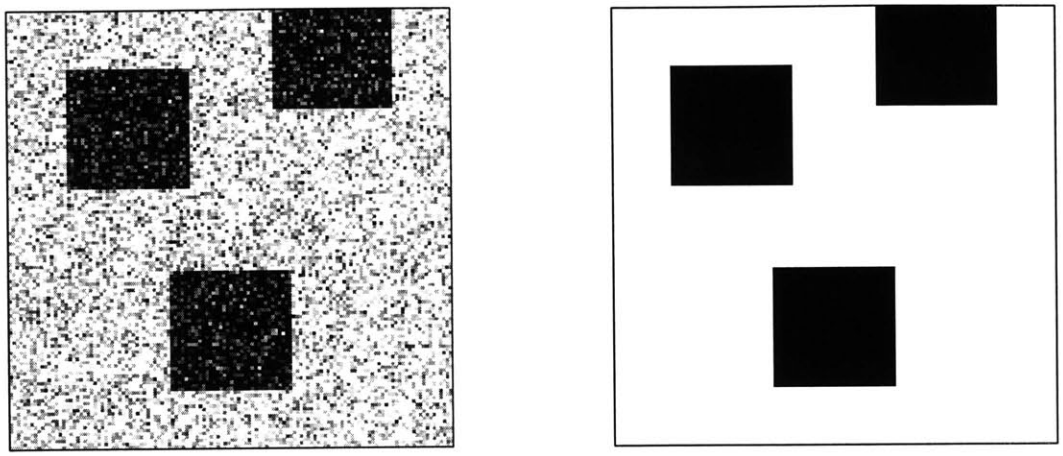

Figure 5-1: Frame number 30 of the test image sequence: noisy frame $u_0(x, y, t)$ on the left and clean frame $u(x, y, t)$ on the right.

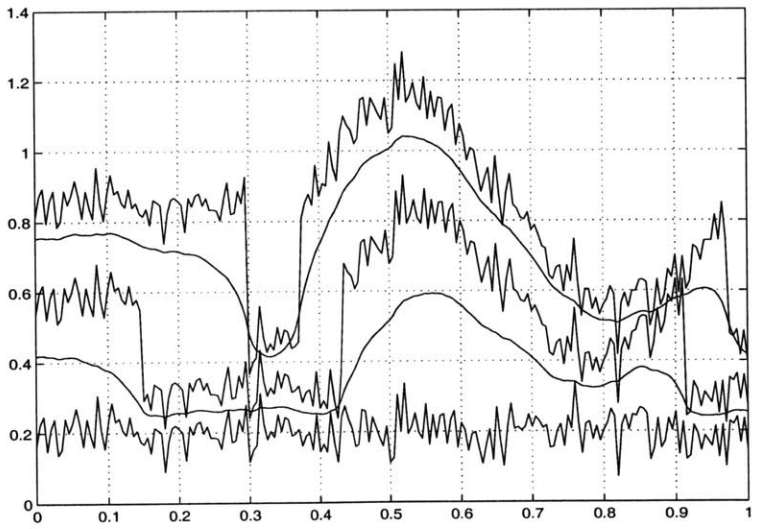

Figure 5-2: Noisy time-varying signal and its estimate from zero initial condition at times $t = 0, 2.5$, and 5.0 sec.

The system therefore satisfies the nonlinear reaction-diffusion equation

$$\hat{u}_t - u_{0,t} = \text{div } h(\nabla \hat{u}) + (u_0 - \hat{u}).$$

The dynamics of $\hat{u}$ now contains $u_{0,t}$ which accounts for the motion compensation.

Let us see how equation (5.1) performs on the same time-varying signal as in Figure 5-2. The noisy signal and its estimate $\hat{u}$ using the same initial condition as previously are shown in Figure 5-3 at times $t = 0, 2.5$, and 5.0 sec. again. We can see that the performance is much better. The edges are kept very well and the noise is removed.

Two-dimensional signals

Similar to the one-dimensional case we will apply the same model we discussed for time-invariant images also to the time-varying case. The only slight modification is that we use a coordinate error feedback again. The enhancement procedure to calculate the estimate $\hat{u}(x, y, t)$ of a noisy time-varying image sequence $u_0(x, y, t)$ is then given by

$$\begin{aligned} \bar{u}_t &= \text{div}(g(|\nabla \hat{u}|)\ \nabla \hat{u}) + (u_0 - \hat{u}) \\ \hat{u} &= \bar{u} + u_0, \end{aligned} \tag{5.2}$$

where $g(s)$ is given by any of the last three entries in Table (2.1). The system therefore satisfies

$$\hat{u}_t = u_{0,t} + \text{div}(g(|\nabla \hat{u}|)\ \nabla \hat{u}) - (\hat{u} - u_0). \tag{5.3}$$

Thus the dynamics of $\hat{u}$ contains $u_{0,t}$, although the actual computation is done using equation (5.2) and hence $u_{0,t}$ is not explicitly used.

The performance of this scheme on the noisy test sequence is shown in Figure 5-4 on the right. Note that the noise is time-invariant, thus only the squares in the sequence move. It is also important to note that this enhancement procedure only works because $u_{0,t}$, which accounts for the motion compensation, is itself not noisy.

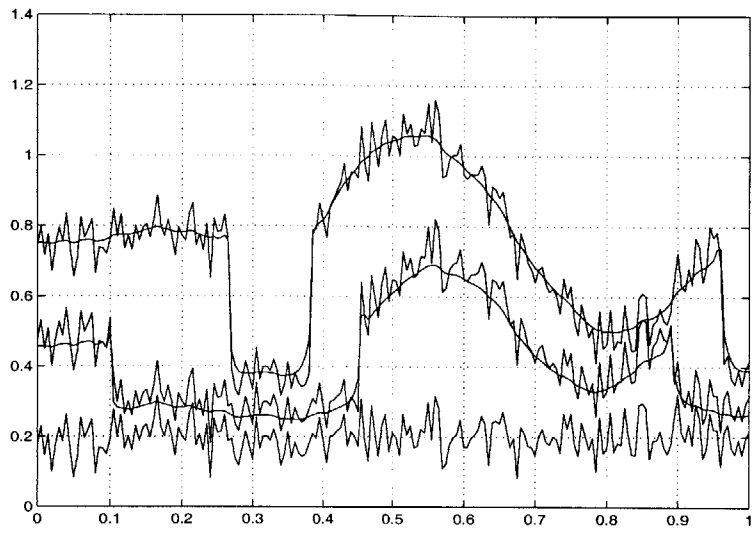

Figure 5-3: Noisy time-varying signal and its estimate $\hat{u}$ from the coordinate error feedback at times $t = 0, 2.5$, and 5.0 sec.

Since the noise does not change in time, the time derivative $\frac{\partial n}{\partial t} = 0$ and $\frac{\partial u_0}{\partial t}$ reduces to

$$\frac{\partial u_0}{\partial t} = \frac{\partial u}{\partial t},$$

which contains no noise at all. However, if the noise changes in time we would get

$$\frac{\partial u_0}{\partial t} = \frac{\partial u}{\partial t} + \frac{\partial n}{\partial t}.$$

A white noise enters the dynamics of $\hat{u}$ through $u_{0,t}$ now and the filtering process does not work anymore.

Finally, consider the special case where an image sequence shows a static scene which is taken from a moving camera whose motion is known. In this case the motion field $u_{0,t}$ stems from that known camera motion. We can replace $u_{0,t}$ with a modeled plant dynamics $\mathbf{v}^T \nabla \hat{u}$, where $\mathbf{v}$ accounts for camera translation and rotation [18].

5.2.2 The time-varying noise case

We will now move on to the case where the noise is assumed to be time-varying. Introducing a slight modification we can still use the model (5.2) proposed earlier

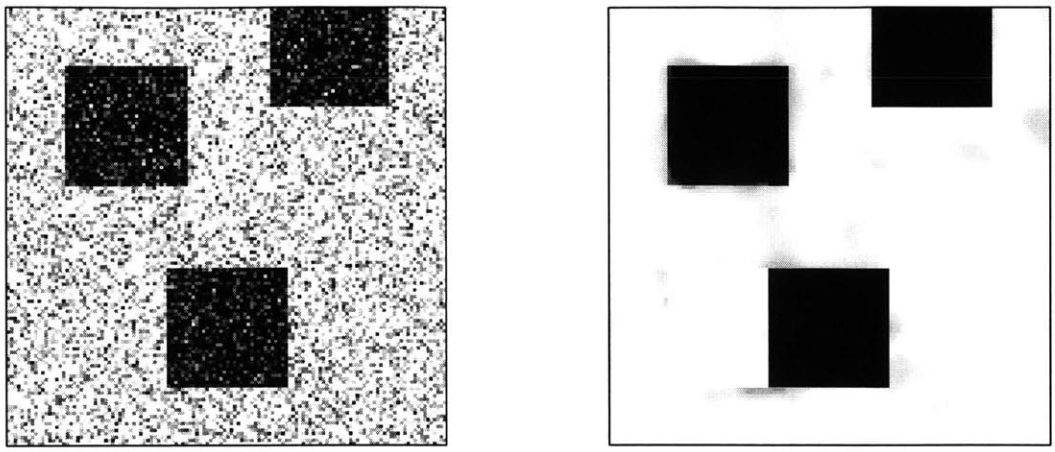

Figure 5-4: Noisy and restored frame 30 of time-varying image sequence with constant noise using coordinate error feedback.

and write

$$\begin{aligned} \bar{u}_t &= \operatorname{div}(g(|\nabla \hat{u}|)\ \nabla \hat{u}) + (u_0 - \hat{u}) \\ \hat{u} &= \bar{u} + K(u_0)\ u_0, \end{aligned} \tag{5.4}$$

where the gain K is a function depending on u_0. The idea is to switch the coordinate error on or off depending on information given in the original image. It is based on the observation that the coordinate error feedback has to be active only at places where important changes actually occur, e.g., at edges of (moving) objects in the image. Thus, in general we will take $K(s) = 1$ for $s \geq \alpha$ and $K(s) = 0$ otherwise, where α is a constant threshold chosen appropriately. Several choices are possible for the gain $K(s)$:

1. $K = K(|\nabla u_0|)$

   We can simply switch on the coordinate error feedback at edges in the original image u_0. This leads to the gain

   $$K(|\nabla u|) = \begin{cases} 1 & \text{if } |\nabla u_0| \geq \alpha \\ 0 & \text{otherwise.} \end{cases} \tag{5.5}$$

   The result of this approach is shown in Figure 5-5. The original noisy image (frame number 30) is plotted on the right and the restored frame is plotted on

the left. Although the background and the squares are restored very well, this method only works for very low noise (in the displayed case the variance of the noise is $\sigma = 0.015$, whereas the difference in gray-value between the objects and the background is 1). Even with low noise there is "false" feedback where the gradient of the noise is higher than the threshold. To be able to deal with higher noise in the image sequence we could base our decision not on the gradient itself but on the estimate of the gradient. In this case K is given by

$$K(|\nabla(G * u_0)|) = \begin{cases} 1 & \text{if } |\nabla(G * u_0)| \geq \alpha \\ 0 & \text{otherwise,} \end{cases} \tag{5.6}$$

where G is a Gaussian function and '$*$' denotes convolution. The restored frame using this choice for K is show in Figure 5-6. The noise in the image sequence has a variance of $\sigma = 0.05$ now and the algorithm gives almost the same result as in the previous case. However, it is obvious that the places where the feedback is turned on are not as exact as in the first case.Furthermore the feedback is also turned on at edges of stationary objects (see the square on the top left), which is not necessary and reduces the quality of the restoration.

2. $K = K(|\frac{\partial}{\partial t}(u_0)|)$

   To prevent feedback at edges of stationary objects we can use

$$K(|\frac{\partial}{\partial t}(u_0)|) = \begin{cases} 1 & \text{if } |\frac{\partial}{\partial t}(u_0)| \geq \alpha \\ 0 & \text{otherwise.} \end{cases} \tag{5.7}$$

   The result of this method is shown in Figure 5-7. The restoration of the squares and the background is very good. Compared to the first idea this scheme has the advantage that the coordinate error feedback is active only at moving edges, and not at stationary ones. Therefore the restoration of the stationary square on the top left is better. But the main problem concerning its applicability in practice remains. The method only works for very low noise ($\sigma = 0.015$) and there is a considerable amount of false feedback. Similarly to the previous case

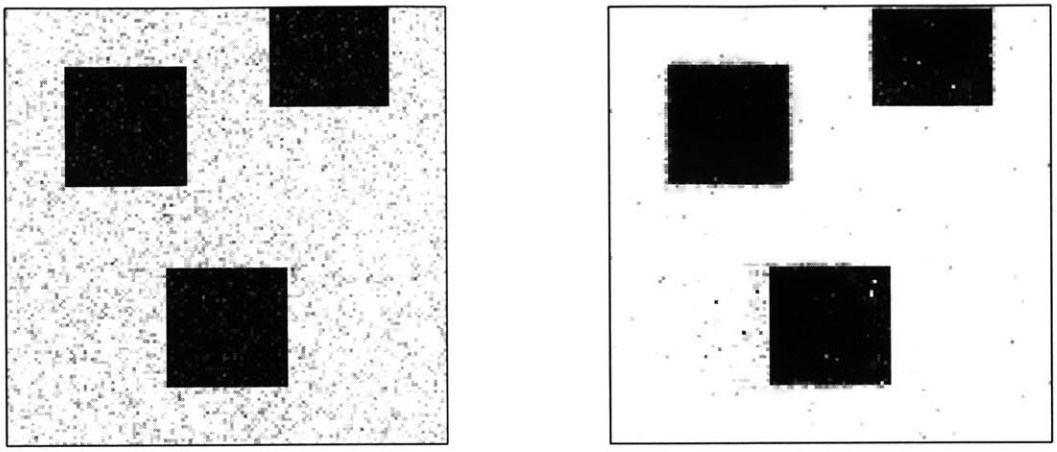

Figure 5-5: Frame number 30 of noisy image sequence on the left and restored frame using K from equation (5.5).

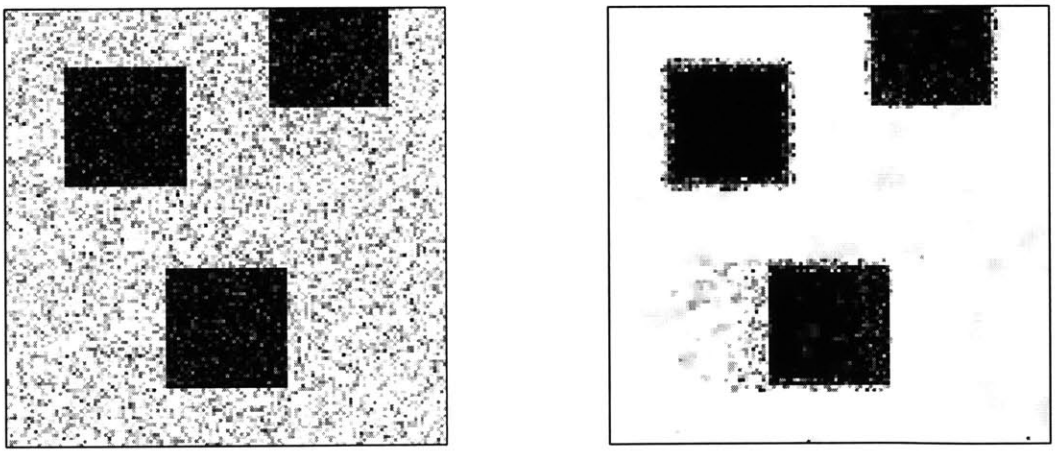

Figure 5-6: Frame number 30 of noisy image sequence on the left and restored frame using K from equation (5.6).

we can therefor use

$$K(|\frac{\partial}{\partial t}(G * u)|) = \begin{cases} 1 & \text{if } |\frac{\partial}{\partial t}(G * u_0)| \geq \alpha \\ 0 & \text{otherwise.} \end{cases} \tag{5.8}$$

Figure 5-8 shows the noisy image with additive noise of variance $\sigma = 0.025$. The restoration of the objects and the background is again very good. The disadvantage of this method is the low accuracy of where to turn on the feedback.

We just saw that a simple extension of the scheme given in equation (5.2) is limited to very low amounts of noise. We will therefore propose a new algorithm now that overcomes those difficulties and gives very good results in the time-varying noise case even in the presence of high noise.

5.3 A Variational Formulation in 3-D

Consider the noisy image sequence shown schematically in Figure 5-9(a). Let us first assume that all N frames are recorded and available for processing. The image sequence is described by equation (1.3) and the noise is white in space as well as in time. Our goal is to find an estimate $\hat{u}(x, y, t)$ of the clean data $u(x, y, t)$ given the noisy observation $u_0(x, y, t)$. Similar to Chapter 2 we can formulate this problem in terms of the minimization problem

$$\inf_{\hat{u}} \left\{ \iiint_{\Omega^+} \left(\frac{1}{2} (\hat{u} - u_0)^2 + \phi(|\nabla^* \hat{u}|) \right) dx\, dy\, dt \right\}, \tag{5.9}$$

where we define $\nabla^* = (\frac{\partial}{\partial x}\ \frac{\partial}{\partial y}\ \frac{\partial}{\partial t})^T$ and ϕ has to satisfy the conditions (1.13) and (2.13) stated for optimal enhancement. The choice of ϕ we will use here is given by equation (4.11). We solve the Euler-Lagrange equation of this minimization problem using the steepest descent method and introduce an artificial evolution paramter τ[1]. The estimate $\hat{u}$ is then given by the steady-state solution of

$$\begin{aligned} \hat{u}_\tau &= \operatorname{div}\left(\frac{\phi'(|\nabla^* \hat{u}|)}{|\nabla^* \hat{u}|} \nabla^* \hat{u} \right) + (u_0 - \hat{u}) \\ \hat{u}(x, y, t, \tau = 0) &= u_0(x, y, t). \end{aligned} \tag{5.10}$$

[1]Note that $\hat{u}$ is already a function of time t and thus a new evolution paramter has to be introduced.

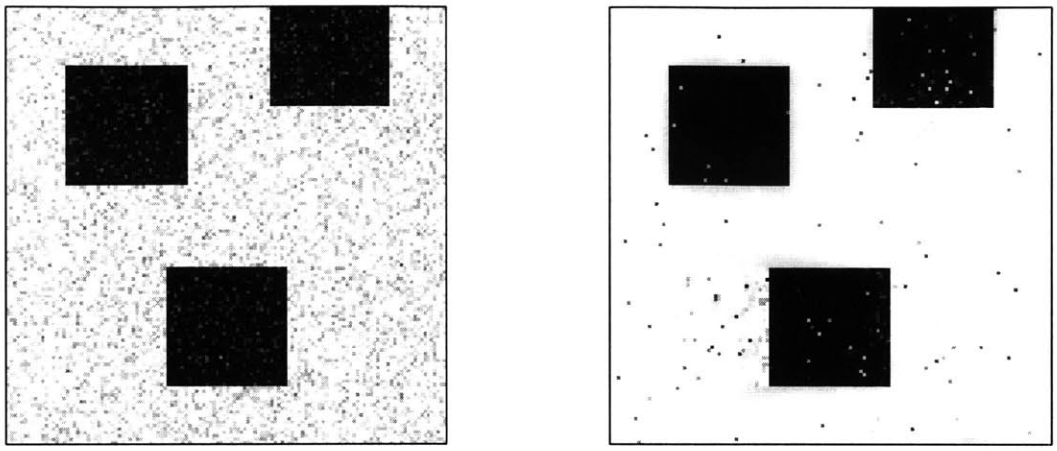

Figure 5-7: Frame number 30 of noisy image sequence on the left and restored frame using K from equation (5.7).

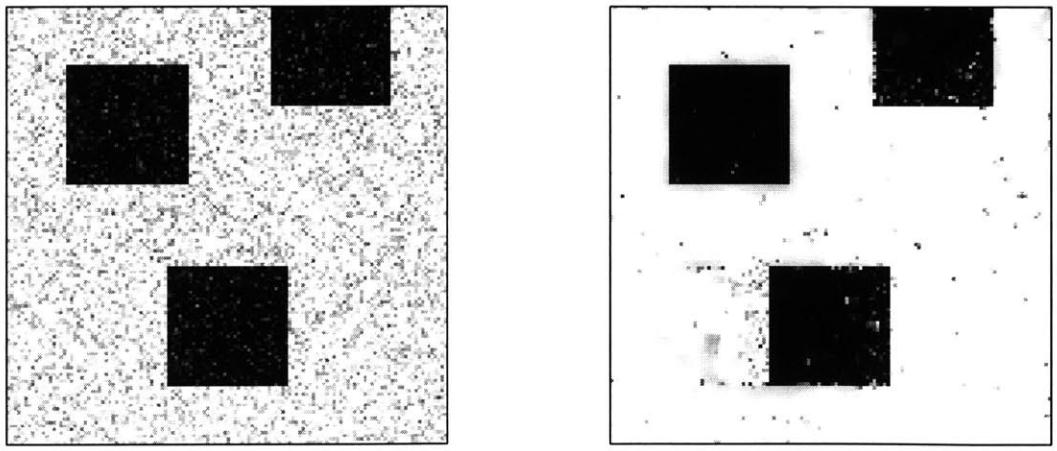

Figure 5-8: Frame number 30 of noisy image sequence on the left and restored frame using K from equation (5.8).

With this approach we do not only keep sharp edges in x and y but also in time t. The original and restored frame number 30 obtained with this method are shown in Figure 5-10 on the left and right, respectively. The noise added to the clean image sequence has a variance of $\sigma = 0.1$. Although the result is very good we should note that the computation can only be done off-line in a post-processing step.

One way to speed up the computation is to consider only 3 frames of the sequence as shown in Figure 5-9(b). Let us assume that the noisy frame $u_{0,n+1}$ is the most recent observation and that we already have a clean estimate $\hat{u}_{n-1}$ of frame $n-1$. We can still use equation (5.10) to solve for the estimate of $u_{0,n}$ and then move on to the next frame. The two neighbouring frames now act as Dirichlet boundary conditions in time. Unfortunately it will take too long to obtain the estimate $\hat{u}_{0,n}$ in practice since $\hat{u}_{0,n}$ is only given by the steady-state solution of equation (5.10).

Before we continue we should note the difference here between a pure 2-D diffusion of $u_{0,n}$ and the 3-D diffusion (5.10). In both cases only $\hat{u}_n$ is changed, u_{n-1} and u_{n+1} are actually not altered at all. However, in the 3-D approach the temporal information is taken into account due to the Dirichlet boundary conditions in time; there is more information on which the estimate $\hat{u}_n$ is based. Furthermore, the convergence of $\hat{u}_n$ is faster due to the Dirichlet boundary conditions, as can be seen in Figure 5-11. Note that although the computational burden is slightly higher for the 3-D diffusion, the solution converges in only half the number of steps of the 2-D diffusion.

Let us now consider a slightly different minimization problem given by

$$\inf_{\hat{u}} \left\{ \iiint_{\Omega^+} \left(\frac{1}{2} (\hat{u} - u_0)^2 + \phi_x(\hat{u}_x) + \phi_y(\hat{u}_y) + \phi_t(\hat{u}_t) \right) dx\, dy\, dt \right\}. \tag{5.11}$$

Solving this problem leads to the Euler-Lagrange equation

$$(\hat{u} - u_0) - \frac{\partial}{\partial x} (\phi_x'(\hat{u}_x)) - \frac{\partial}{\partial y} (\phi_y'(\hat{u}_y)) - \frac{\partial}{\partial t} (\phi_t'(\hat{u}_t)) = 0, \tag{5.12}$$

which can be rewritten as

$$(\hat{u} - u_0) - \operatorname{div}(\boldsymbol{\phi}'(\nabla \hat{u})) = 0,$$

where $\boldsymbol{\phi}'(\nabla \hat{u})$ is defined as

$$\boldsymbol{\phi}'(\nabla \hat{u}) = [\phi_x'(\hat{u}_x) \quad \phi_y'(\hat{u}_y) \quad \phi_t'(\hat{u}_t)]^T.$$

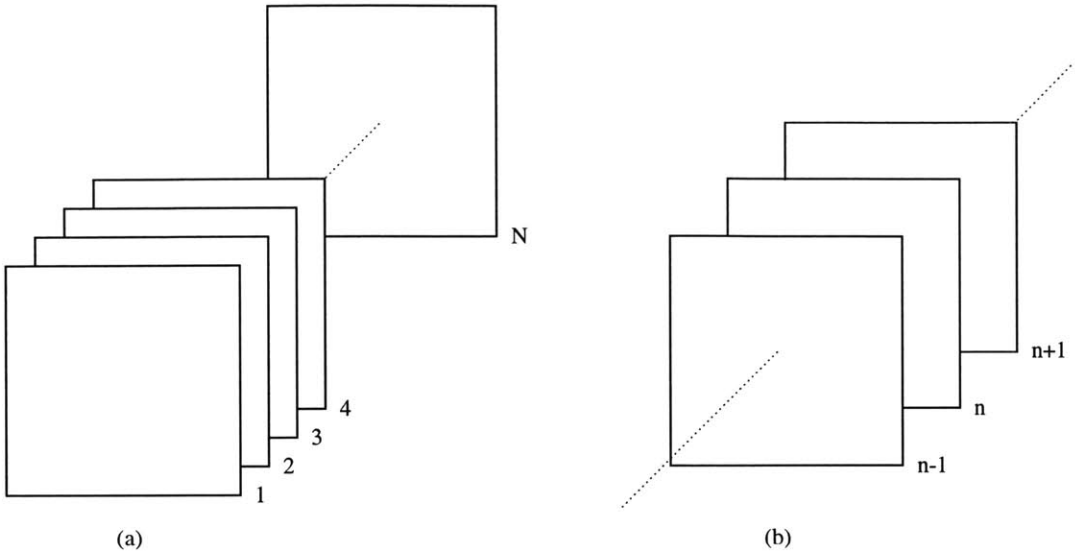

Figure 5-9: (a) Image Sequence. (b) Frames u_{n-1}, u_n, and u_{n+1}.

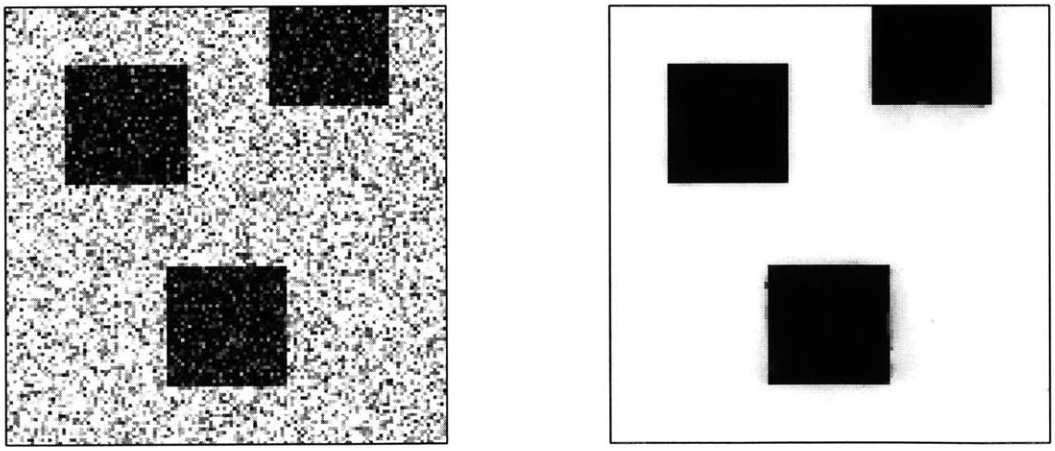

Figure 5-10: Noisy frame u_0 on the left and estimate $\hat{u}$ obtained from a three-dimensional nonlinear diffusion.

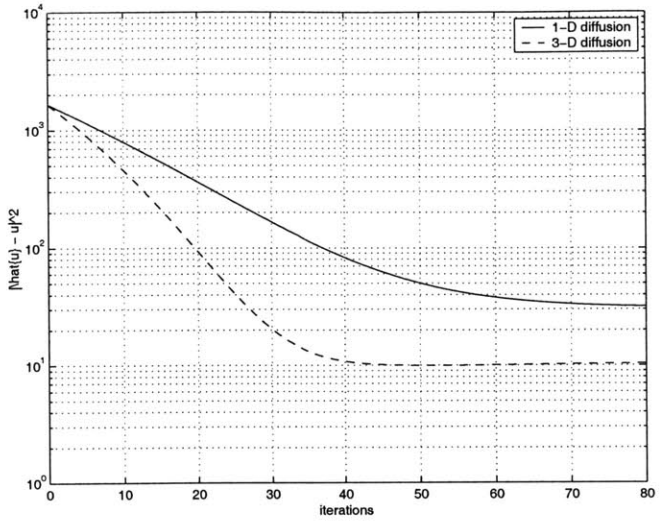

Figure 5-11: Comparison of convergence rates for 1-D and 3-D diffusion.

The main difference between the two minimization problems is that all directions are treated separately in the second one. This has the advantage that the corresponding Euler-Lagrange equation decouples in x, y, and t. To analyze this equation we first neglect the two space dimensions and only look at the optimization in time t. From equation (5.12) we know that the solution of this optimization problem is given by the steady-state solution of

$$\frac{\partial \hat{u}}{\partial \tau}(t,\tau) = \frac{\partial}{\partial t}\left(h\left(\frac{\partial \hat{u}}{\partial t}\right)\right) + (u_0 - \hat{u}), \tag{5.13}$$

where $h = \phi_t'$. The discrete form of this equation is given by

$$\hat{u}_n^{m+1} = \hat{u}_n^m + \frac{\Delta\tau}{\Delta t^2}\left(\mathrm{h}^+(u_{n+1} - \hat{u}_n^m) - \mathrm{h}^-(\hat{u}_n^m - u_{n-1})\right) + \Delta\tau\,(u_{0,n} - \hat{u}_n^m), \tag{5.14}$$

where the superscript m denotes the iteration number, and u_{n-1} and u_{n+1} are the previous and next frame, respectively. Note that the functions h^+ and h^- are equivalent and differ in their argument only. To simplify the subsequent discussion we will now use a pure saturation $\mathrm{h}(s) = \mathrm{sat}(\frac{s}{\theta})$ instead of a smooth one. It is actually possible to find the steady-state solution of equation (5.14) in this special case. This is most evident by means of the example shown in Figure 5-12. The plot shows a pixel of frame $n-1$ and $n+1$ and their corresponding grayvalues u_{n-1} and u_{n+1}. The noisy

observation $\hat{u}_n = u_{0,n}$ can lie either above u_{n+1} (e.g. at position 1), between u_{n-1} and u_{n+1} (position 3) or below u_{n-1} (position 5). Neglecting the feedback term in equation (5.14) for a moment let us assume first that $u_{n,0}$ lies at position 1. In this case $u_{n+1} - \hat{u}_n$ is negative and therefore $\mathrm{h}^+(u_{n+1} - \hat{u}_n) = -1$ is negative. On the other hand $\hat{u}_n - u_{n-1}$ is positive and so is $\mathrm{h}^-(\hat{u}_n - u_{n-1}) = +1$. Since h^- is subtracted from h^+ the absolute value of $\hat{u}_n$ will decrease by the amount $\frac{2\Delta\tau}{\Delta t^2}$. As soon as the pixel at position 1 enters the boundary layer at $(u_{n+1} + \theta)$, h^+ will decrease, whereas h^- will still be $+1$. If $\hat{u}_n = u_{n+1}$ it turns out that $\mathrm{h}^+ = 0$, and if $\hat{u}_n = (u_{n+1} - \theta)$ (position 2) we have $\mathrm{h}^+ = 1$. At this point $\mathrm{h}^+ = \mathrm{h}^-$ and $\hat{u}_n$ will remain constant. Note that $\hat{u}_n$ will remain constant anywhere in between the two boundary layers, since $\mathrm{h}^+ = \mathrm{h}^-$ holds. It is straightforward to do a similar analysis if $\hat{u}_n$ is initially at position 5. In this case $\hat{u}_n$ will reach its steady-state at position 4. Thus we can summarize the following:

- If $u_{0,n} \geq u_{n+1} - \theta$ then $\hat{u}_n \to u_{n+1} - \theta$;
- If $u_{0,n} \leq u_{n-1} + \theta$ then $\hat{u}_n \to u_{n-1} + \theta$;
- If $u_{n-1} + \theta < u_{0,n} < u_{n+1} - \theta$ then $\hat{u}_n = u_{0,n}$.

The same results hold for $u_{n-1} > u_{n+1}$ by simply switching the roles of u_{n-1} and u_{n+1}.

So far we neglected the feedback term $(u_{0,n} - \hat{u}_n)$ in our analysis. So let us check how this term influences the solution. Within the boundary layer the feedback term will be 0, since $\hat{u}_n = u_{0,n}$. If $u_{0.n}$ is initially at position 1, the feedback term will try to pull $\hat{u}_n$ back up. Thus $\hat{u}_n$ will not reach its steady-state at positon 2, but slightly above[2]. A similar result holds if $u_{0,n}$ is originally at position 5.

The relation between the input value $u_{0,n}$ and the output $\hat{u}_n$ can therfore be written as $\hat{u}_n = f(u_{0,n}, u_{n-1}, u_{n+1})$. We just saw that the influence of the feedback term is very small. To simplify the problem we will therefore neglect the feedback

[2]Note that $0 \leq u \leq 1$ and therefore $(u_{0,n} - \hat{u}_n) \leq 1$. However, $-1 \leq \mathrm{h} \leq 1$ and therefore $\hat{u}_n$ will lie between $u_{n+1} - \theta$ and u_{n+1}.

term and also set $\theta = 0$. Note that we can then write this mapping in closed form as

$$\hat{u}_n = \frac{1}{2} |(u_{n+1} - u_{n-1})| \text{ sat} \left(\frac{u_{0,n} - \frac{1}{2}(u_{n+1} + u_{n-1})}{\frac{1}{2}|(u_{n+1} - u_{n-1})|} \right) + \frac{1}{2}(u_{n+1} + u_{n-1}). \quad (5.15)$$

The shape of $f(u_{0,n}, u_{n-1}, u_{n+1})$ is shown in Figure 5-13.

A well known method in numerical analysis is the alternating direction implicit (ADI) method. The main idea of this method is to solve a multidimensional equation in each direction seperately, thus reducing the bandwidth of the resulting system of equations. In the second minimization problem we stated that the system is decoupled in each direction, thus using ADI methods is a natural thing to do. We also obtained an approximate solution to the one-dimensional optimization problem in time given by equation (5.15). We therefore propose to solve equation (5.12) in the following way: first do the mapping (5.15) in time and then do an equivalent mapping in x and y. If low-frequency noise remains in the image, we can compare every pixel not only to its direct neighbour in x and y, but also to their second next neighbours, thus increasing the smoothing scale of the algorithm.

Before showing results of this approach let us write down the algorithm in a general form. The discrete version of u is defined by

$$u(i, j, n) = u_{i,j,n} = u(i\Delta x, j\Delta y, n\Delta t),$$

where Δt is the temporal stepsize and Δx and Δy are the spatial stepsizes. We will also need the following definitions

$$\begin{aligned}
\delta_x^{i,j,n} &= \{u(i-1, j, n), u(i+1, j, n)\} \\
\delta_y^{i,j,n} &= \{u(i, j-1, n), u(i, j+1, n)\} \\
\delta_t^{i,j,n} &= \{u(i, j, n-1), u(i, j, n+1)\} \\
\delta_{2x}^{i,j,n} &= \{u(i-2, j, n), u(i+2, j, n)\} \\
\delta_{2y}^{i,j,n} &= \{u(i, j-2, n), u(i, j+2, n)\}.
\end{aligned}$$

Furthermore we use the following notation

$$\begin{aligned}
\Delta\delta_x^{i,j,n} &= (u(i+1, j, n) - u(i-1, j, n)) \\
\bar{\delta}_x^{i,j,n} &= \frac{1}{2}(u(i-1, j, n) + u(i+1, j, n)).
\end{aligned}$$

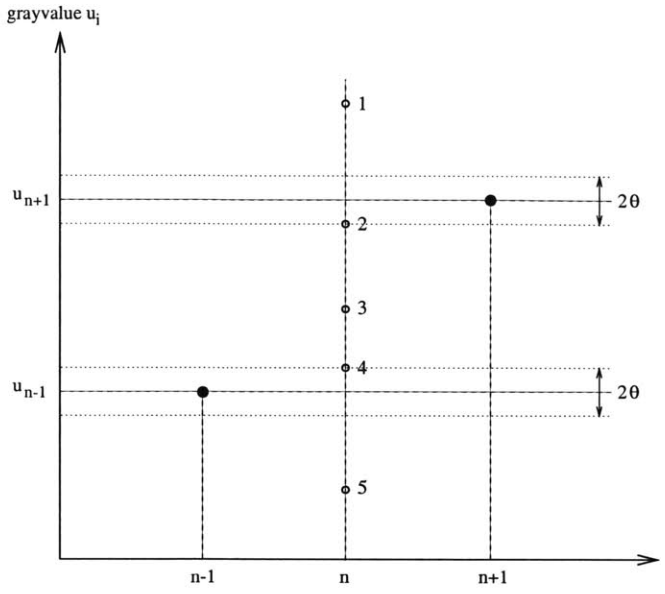

Figure 5-12: Mapping of $u_{0,n}$.

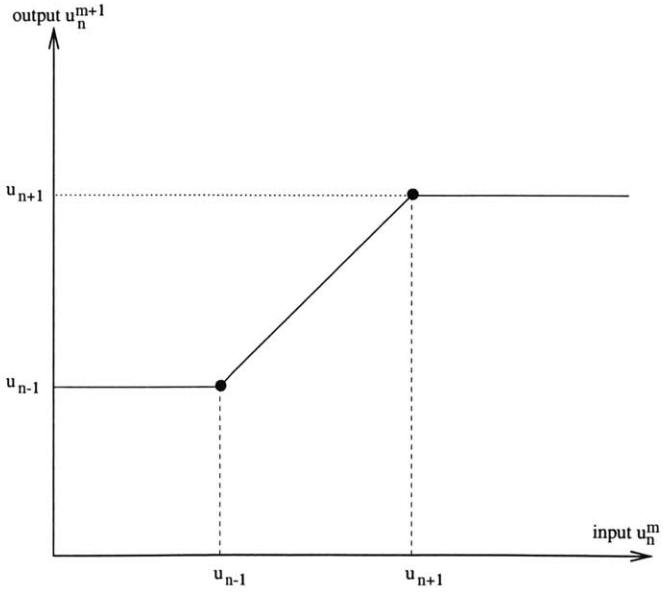

Figure 5-13: Mapping of $u_{0,n}$ to $\hat{u}_n$.

A similar notation of $\Delta\delta_k^{i,j,n}$ and $\bar{\delta}_k^{i,j,n}$ applies to $k = y,\, t,\, 2x,\, 2y$. The relation between the input and the output including the boundary layer ($\theta > 0$) can then be written as

$$f_k(u_{i,j,n}) = \begin{cases} (\max \delta_k^{i,j,n} - \theta) & \text{if } u_{i,j,n} \geq (\delta_k^{i,j,n} - \theta) \ \vee \ |\Delta\delta_k^{i,j,n}| \geq 2\theta \\ (\min \delta_k^{i,j,n} + \theta) & \text{if } u_{i,j,n} \leq (\delta_k^{i,j,n} + \theta) \ \vee \ |\Delta\delta_k^{i,j,n}| \geq 2\theta \\ \bar{\delta}_k^{i,j,n} & \text{if } u_{i,j,n} \lesseqgtr (\delta_k^{i,j,n} \mp \theta) \ \vee \ |\Delta\delta_k^{i,j,n}| < 2\theta \\ u_{i,j,n} & \text{otherwise,} \end{cases} \tag{5.16}$$

where we define $u_{i,j,n} \geq \delta_k^{i,j,n}$ to be true only if it holds for either entry of $\delta_k^{i,j,n}$ and correspondingly for the other inequalities. If we let $\theta = 0$ equation (5.16) reduces to

$$f_k(u_{i,j,k}) = \begin{cases} \max \delta_k^{i,j,n} & \text{if } u_{i,j,n} \geq \delta_k^{i,j,n} \\ \min \delta_k^{i,j,n} & \text{if } u_{i,j,n} \leq \delta_k^{i,j,n} \\ u_{i,j,n} & \text{otherwise.} \end{cases} \tag{5.17}$$

Similar to equation (5.15) this can be written in closed form as

$$f_k(u_{i,j,k}) = \frac{1}{2}\, |\Delta\delta_k^{i,j,n}|\, \text{sat}\left(\frac{u_{i,j,k} - \bar{\delta}_k^{i,j,n}}{\frac{1}{2}\, |\Delta\delta_k^{i,j,n}|} \right) + \bar{\delta}_k^{i,j,n}. \tag{5.18}$$

The estimate $\hat{u}_{i,j,n}$ of the noisy observation $u_{0,i,j,n}$ is computed according to the following 5 steps:

- $u_{i,j,n}^1 = f_t(u_{0,i,j,n})$
- $u_{i,j,n}^2 = f_x(u_{i,j,n}^1)$
- $u_{i,j,n}^3 = f_y(u_{i,j,n}^2)$
- $u_{i,j,n}^4 = f_{2x}(u_{i,j,n}^3)$
- $\hat{u}_{i,j,n} = f_{2y}(u_{i,j,n}^4)$.

It is important to note that the proposed algorithm satisfies the following Min-Max-Criterion

$$\min_{i,j,n}\ u_0(i,j,n) \ \leq \ \hat{u}(i,j,n) \ \leq \ \max_{i,j,n}\ u_0(i,j,n). \tag{5.19}$$

The result obtained with the new algorithm is shown in Figure 5-14. The noisy frame is shown on the left and the restored one on the right. The computation to process one frame takes only 0.12 sec. and the algorithm therefore runs with a speed of approximately 8 frames per second. Figure 5-15 shows a comparison between the convergence of the 2-D diffusion, the 3-D diffusion and the new algorithm on the left and a zoom on the first 10 iterations on the right. The plot on the right shows that the proposed algorithm basically convergences after only two steps (first step: $t-x-y$, second step: $2x-2y$). Note that doing one step of the new algorithm takes the same computation time as one step of the discrete 3-D diffusion. The proposed scheme is therefore approximately 20 times faster than the explicit 3-D diffusion algorithm using the discretization shown in section 4.5.

Finally, Figure 5-16 summarizes the results obtained from the 2-D diffusion (top right), the 3-D diffusion (bottom left), and the new algorithm (bottom right). The signal-to-noise ratio of the original noisy frame is $\mathrm{SNR}(u/u_0) = 8.5$, the one of the result using the 2-D diffusion is $\mathrm{SNR}(u/\hat{u}) = 36.8$, the 3-D diffusion $\mathrm{SNR}(u/\hat{u}) = 43.8$ and the new algorithm $\mathrm{SNR}(u/\hat{u}) = 39.2$. The performance of the proposed scheme is visually as good as the full 3-D diffusion and even better than a pure 2-D diffusion.

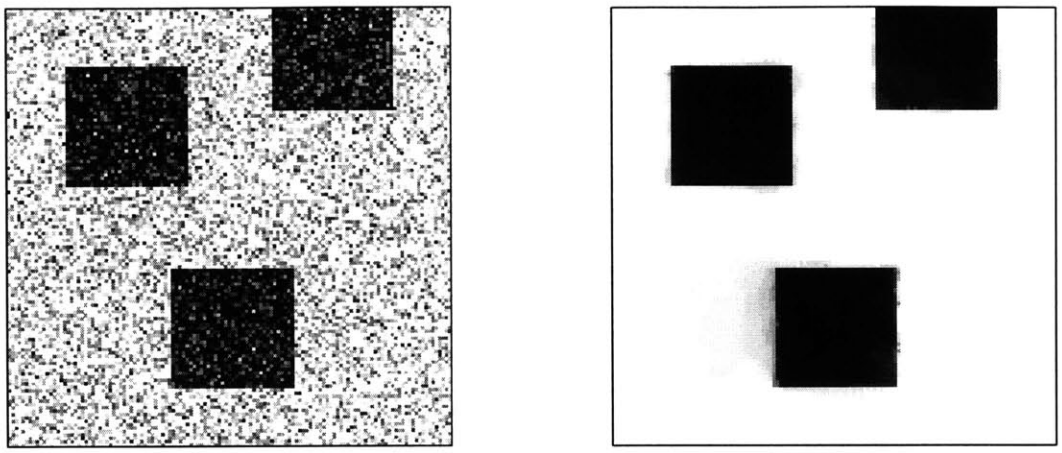

Figure 5-14: Noisy frame u_0 on the left and estimate $\hat{u}$ obtained from the new proposed algorithm.

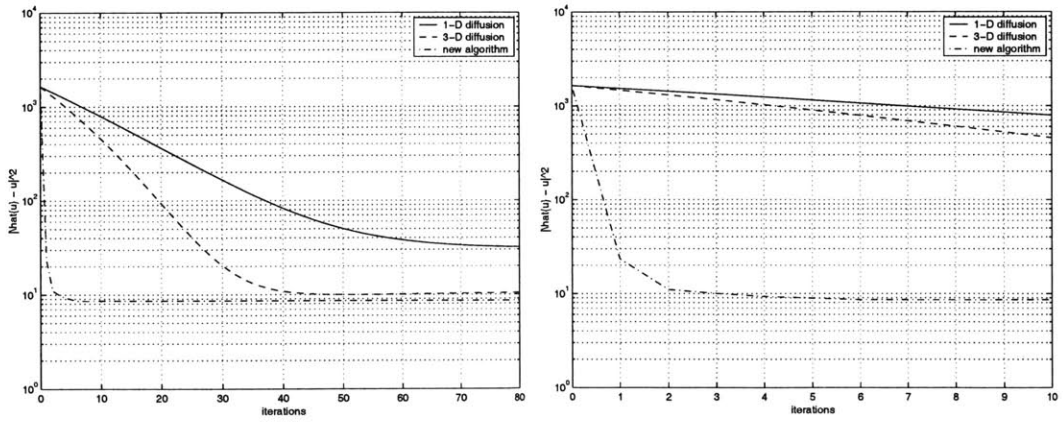

Figure 5-15: Left: Comparison of convergence rates. Right: Zoom on the first 10 iterations.

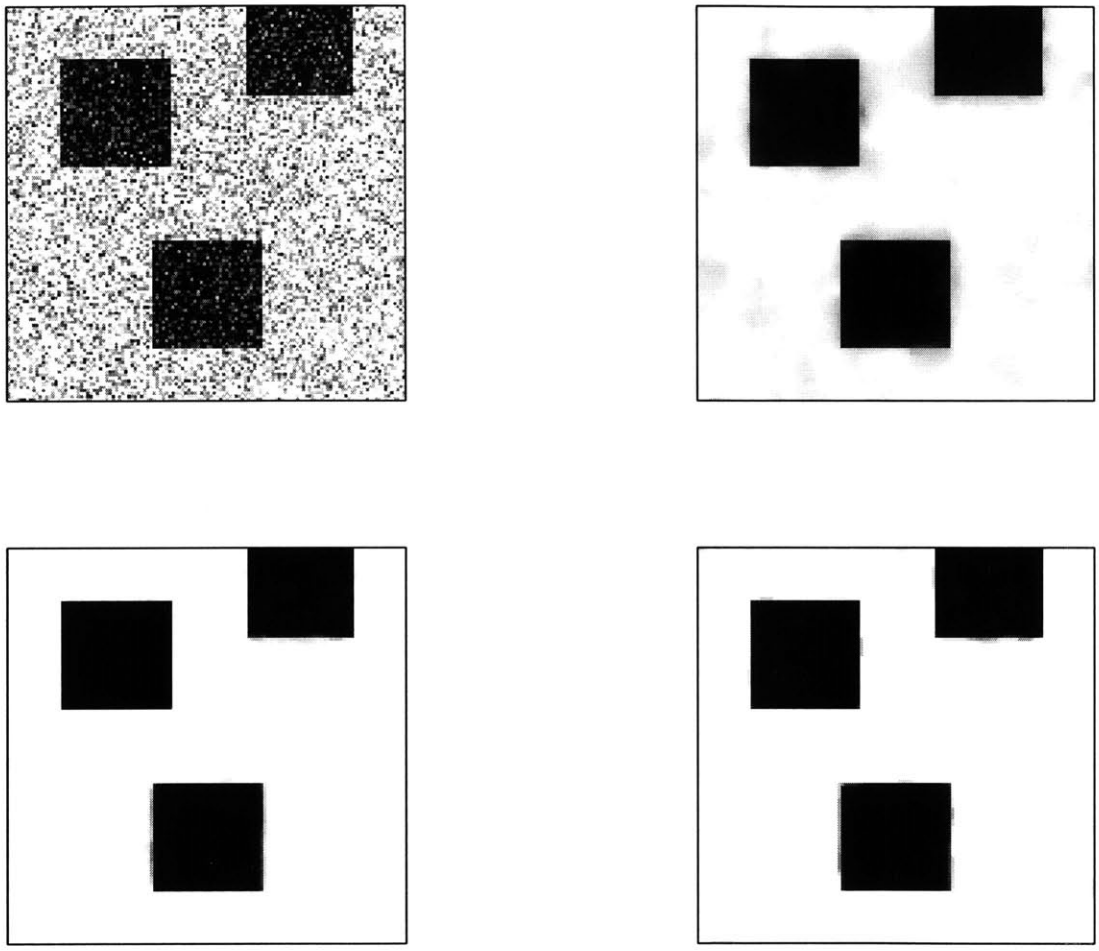

Figure 5-16: Comparison of results of 2-D, 3-D diffusion and new algorithm.

Chapter 6

Conclusions

6.1 Summary

In this study, we sought to give an overview of the application of partial differential equations in image processing and extend the results existing for the enhancement of static images to the enhancement of time-varying image sequences.

The diffusion equation was the first PDE used for image enhancement because its solution is equivalent to the result obtained from filtering by Gaussian convolution. We saw that using nonlinearities in the divergence operator substantially improved the results obtained on the test image. It was also shown that nonlinear diffusion equations can also be derived from a variational formulation, thus giving a mathematical justification for their use in image enhancement.

In the context of curvature flows the image was interpreted as a collection of iso-intensity contours that evolved under their curvature. Level set methods were introduced to make ideas clearer. They are a convenient computational tool for evolving curves and can also be used for other image processing tasks, such as geodesic active contours or snakes. The main advantage of curvature flows was that no smoothing took place across the level lines, thus keeping boundaries sharp. Shock filters achieve the aim of deblurring or rebuilding the edges best, but their application is limited in the presence of noise. They can be used to sharpen the edges in the last stage of processing the image after most of the noise is removed.

A result obtained from extending contraction analysis to the stability analysis of a certain class of nonlinear reaction-diffusion equations was given. We discussed the conditions that have to be satisfied by the nonlinearity in the divergence operator for the system to be contracting. Results were shown using a smooth saturation function as the nolinearity. This choice reduces to a curvature flow for high gradients and to an isotropic diffusion for low gradients, thus giving a very good performance in keeping the edges and smoothing isotropically within regions of uniform brightness.

Finally, we extended the results obtained for the enhancement of static images to time-varying image sequences. A simple extension using a coordinate error feedback proved to be useful only in the time-invariant noise case. We also showed a way to circumvent this problem which was restricted to cases with only low amounts of noise. A new algorithm for image sequence enhancement with time-varying noise was then derived. The idea of the algorithm is based on ADI methods in numerical analysis that are used to obtain a fast approximate solution to the three-dimensional variatonal problem. A maximum principle was given and results obtained on synthetic image sequences were shown.

6.2 Conclusion

Contraction analysis is a very convenient tool to analyze global or local stability properties of nonlinear systems. There is a wide variety of applications, especially in controller and observer designs of mechanical and chemical systems. In this study we considered its application to image processing by using results obtained for the analysis of nonlinear reaction-diffusion processes. The main advantage of applying contraction analysis in this field is that global exponential convergence rates can be quantified.

Algorithms for image enhancement using partial differential equations are very well developed and their performance is very good. However, the extension of those algorithms to the enhancement of time-varying image sequences in real-time is missing. We proposed a fast and recursive algorithm that has the capability to run in

real-time. The results on synthetic data presented in this thesis are very promising.

6.3 Future Work

Further work on this topic would most certainly involve a profound theoretical analysis of the algorithm for real-time image enhancement presented here. Numerical aspects like multigrid methods have to be investigated and a computer code has to be developed to further speed up the computation.

We also believe that the performance could be further improved using motion compensation. The velocity vector field could be estimated and thus improve the time filtering capabilites of the system.

Finally, since all computations are done using discrete equations (or better the discrete approximations of continuous equations) it would be interesting to study and derive results for these equations directly.

Bibliography

[1] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel. Axioms and fundamental equations of image processing. *Archive for Rational Mechanics and Analysis*, 16(9):200–257, 1993.

[2] L. Alvarez, P.-L. Lions, and J.-M. Morel. Image selective smoothing and edge detection by nonlinear diffusion II. *SIAM Journal on Numerical Analysis*, 29(3):845–866, 1992.

[3] G. Aubert, M. Barlaud, L. Blanc-Feraud, and P. Charbonnier. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 5(12), 1997.

[4] G. Aubert and L. Vese. A variational method in image recovery. *SIAM Journal on Numerical Analysis*, 34(5):1948–1979, 1997.

[5] J. Babaud, A.P. Witkin, M. Baudin, and R.O. Duda. Uniqueness of the gaussian kernel for scale-space filtering. *IEEE Transactions PAMI*, 8(1):26–33, 1986.

[6] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. Journal of Computer Vision*, 22(1):61–79, 1997.

[7] A. Chambolle and P.-L. Lions. Image recovery via total variation minimization and related problems. *Research Note 95-09, CEREMADE, Univ. Paris-Dauphine, France*, 1995.

[8] R. Deriche and O. Faugeras. Les edp en traitment des images et vision par ordinateur. *Traitement du Signal*, 13(6), 1996.

[9] L.C. Evans and J. Spruck. Motion of level sets by mean curvature I. *J. Differential Geometry*, 33:635–681, 1991.

[10] L. Florack. *Image Structure.* Kluwer, 1997.

[11] L. Florack. Pseudo-linear scale-space theory. *Int. Journal of Computer Vision*, 31(2/3), 1999.

[12] L. Florack, A.H. Salden, B.M. ter Haar Romeny, J.J. Koenderink, and M.A. Viergever. Nonlinear scale-space. *Image and Vision Computing*, 13(4), 1995.

[13] M. Gage and R.S. Hamilton. The heat equation shrinking convex plane curves. *J. Differential Geometry*, 23:69–96, 1986.

[14] M.E. Gage. Curve shortening makes convex curves circular. *Inventions mathematicae*, 76:357–364, 1984.

[15] M.A. Grayson. The heat equation shrinks embedded plane curves to round points. *J. Differential Geometry*, 26:285–314, 1987.

[16] P.J. Green. Bayesian reconstruction from emission tomography data using a modified em algorithm. *IEEE Trans. Med. Imaging*, 9:84–93, 1990.

[17] T. Hebert and R. Leahy. A generalized em algorithm for 3-d bayesian reconstruction from poisson data using gibbs priors. *IEEE Trans. Med. Imaging*, 8:194–202, 1989.

[18] B.K.P. Horn. *Robot Vision.* MIT Press, 1986.

[19] M. Kass, A. Witking, and D. Terzopoulus. Snakes: Active contour models. *Int. Journal of Computer Vision*, pages 321–331, 1988.

[20] B.B. Kimia and K. Siddiqi. Geometric heat equation and nonlinear diffusion of shapes and images. *Computer Vision and Image Understanding*, 64(3):305–322, 1996.

[21] J.J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[22] P.D. Lax. *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves.* SIAM. 1973.

[23] R.J. LeVeque. *Numerical Methods for Conservation Laws.* Lecture Notes in Mathematics, ETH Zürich. Birkhäuser, 1992.

[24] W. Lohmiller and J.J.E. Slotine. On contraction analysis for nonlinear systems. *Automatica*, 34(6), 1998.

[25] W. Lohmiller and J.J.E. Slotine. Stability analysis and observer design for nonlinear reaction-diffusion processes. In H. Nijmeijer and T. Fossen, editors, *New Trends in Nonlinear Observer Design.* Springer Verlag, 1999.

[26] W. Lohmiller and J.J.E. Slotine. Contraction analysis of nonlinear reaction-diffusion processes. *European Control Conference*, Karlsruhe, 1999.

[27] D. Marr and E. Hildreth. Theory of edge detection. *Proc. Roy. Soc. Lond.*, B(207):187–217, 1980.

[28] N.K. Nordstroem. Biased anisotropic diffusion: a unified regularization and diffusion approach to edge detection. *Image and Vision Computing*, 8(4):318–327, 1990.

[29] S. Osher and L.I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis*, 27(4):919–940, 1990.

[30] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on a Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79(12), 1985.

[31] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions PAMI*, 12(7):629–639, 1990.

[32] L.I. Rudin. *Images, Numerical Analysis of Singularities and Shock Filters.* PhD thesis, California Institute of Technology, 1987.

[33] L.I. Rudin and S. Osher. Total variation based image resoration with free local constraints. *IEEE*, pages 31–35, 1994.

[34] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[35] G. Sapiro and A. Tannenbaum. Affine invariant scale-space. *Int. Journal of Computer Vision*, 11(1):25–44, 1993.

[36] G. Sapiro and A. Tannenbaum. On affine plane curve evolution. *Journal of Funtional Analysis*, 119:79–120, 1994.

[37] G. Sapiro and A. Tannenbaum. Area and length preserving geometric invariant scale-space. *IEEE Transactions PAMI*, 17(1):67–72, 1995.

[38] J.A. Sethian. *Level Set Methods and Fast Marching Methods.* Cambridge University Press, 1999.

[39] B.M. ter Haar Romeny (ed.). *Geometry-Driven Diffusion in Computer Vision.* Kluwer, 1994.

[40] V. Torre and T.A. Poggio. On edge detection. *IEEE Transactions PAMI*, 8(2):147–163, 1986.

[41] R. van den Boomgaard. The morphological equivalent of the gauss convolution. *Nieuw Archief voor Wiskunde*, 10(3):219–236, 1992.

[42] R. van den Boomgaard and A. Smeulders. The morphological structure of images: The differential equations of morphological scale-space. *IEEE Transactions PAMI*, 16(11):1101–1113, 1994.

[43] A.P. Witkin. Scale-space filtering. In *Proc. of IJCAI*, pages 1019–1021, Karlsruhe, 1983.