P

Pictures and graphs are in color and therefore
will not scan or reproduce well.

August, 1999

LIDS- TH 2461

Research Supported By:

AFOSR Grant #F49620-96-1-0028
Boston University - ODDR & E. MURI
Subcontract #GC123919NGD
AFOSR Grant #F49620-98-1-0349
ONR Grant #N00014-91-J-1004

Nonlinear Scale Space Analysis in Image Processing

Pollak, Ilya

Nonlinear Scale Space Analysis in Image Processing

by

Ilya Pollak

B.S. and M.Eng, Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 1995

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

August, 1999

Signature of Author: ______________________________
Dept. of Electrical Engineering and Computer Science
August 27, 1999

Certified by: ______________________________
Alan S. Willsky
Professor of EECS
Thesis Supervisor

Certified by: ______________________________
Hamid Krim
Professor of ECE, North Carolina State University,
Thesis Supervisor

Accepted by: ______________________________
Arthur C. Smith
Professor of EECS
Chair, Committee for Graduate Students

Nonlinear Scale Space Analysis in Image Processing

by Ilya Pollak
ipollak@alum.mit.edu

Submitted to the Department of Electrical Engineering
and Computer Science on August 27, 1999
in Partial Fulfillment of the Requirements for the Degree
of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

The objective of this work is to develop and analyze robust and fast image segmentation algorithms. They must be robust to pervasive, large-amplitude noise, which cannot be well characterized in terms of probabilistic distributions. This is because the applications of interest include synthetic aperture radar (SAR) segmentation in which speckle noise is a well-known problem that has defeated many algorithms. The methods must also be robust to blur, because many imaging techniques result in smoothed images. For example, SAR image formation has a natural blur associated with it, due to the finite aperture used in forming the image. We introduce a family of first-order multi-dimensional ordinary differential equations with discontinuous right-hand sides and demonstrate their applicability to segmenting both scalar-valued and vector-valued images, as well as images taking values on a circle. An equation belonging to this family is an inverse diffusion everywhere except at local extrema, where some stabilization is introduced. For this reason, we call these equations "stabilized inverse diffusion equations" ("SIDEs"). Existence and uniqueness of solutions, as well as stability, are proven for SIDEs. A SIDE in one spatial dimension may be interpreted as a limiting case of a semi-discretized Perona-Malik equation [49,50], which, in turn, was proposed in order to overcome certain shortcomings of Gaussian scale spaces [72]. These existing techniques are reviewed in a background chapter. SIDEs are then described and experimentally shown to suppress noise while sharpening edges present in the input image. Their application to the detection of abrupt changes in 1-D signals is also demonstrated. It is shown that a version of the SIDEs optimally solves certain change detection problems. Its relations to the Mumford-Shah functional [44] and to linear programming are discussed. Theoretical performance analysis is carried out, and a fast implementation of the algorithm is described.

Thesis Supervisor: Alan S. Willsky
Title: Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Hamid Krim
Title: Assistant Professor of Electrical and Computer Engineering, North Carolina State University

This thesis is dedicated to the memory of my mother, Tamara B. Skomorovskaya (1935-1997).

Acknowledgments

After spending eight years at MIT, all in the same group and in the same office, I have the dubious distinction of having been in the Stochastic Systems Group (SSG) longer than any current (and, I suspect, past) member except our leader Alan Willsky[1]. I have seen many people go through the group: dozens of graduate students, six secretaries, numerous post-docs, research scientists, and visitors—sufficient to staff an Electrical Engineering department of a medium-size college, and still have enough administrative assistants left over to sustain "Murphy Brown" for at least a couple of seasons.

I have seen two people engaged in a rubber-band fight; they are now a quantitative analyst at a large financial company and an assistant professor at a top-ranked university. I remember an attempted coup d'etat in the group when two other people (also now assistant professors), using the fact that Alan was traveling, decided to take a shot at the sacred tradition of naming the computer equipment after fine wines, by calling our new printer "Gallo". Luckily, Alan's speedy and decisive actions upon his return restored the order: the printer was re-named "Chenas". Yet another member of the group (also an assistant professor now) once surprised me with his masterful use of a few Russian choice words; it turned out that he learned those especially for me, using his five-language dictionary of obscenities. Even more entertaining is the ongoing feud between two current group members: actions taken have included sealing the opponent inside a room, putting something interesting onto his desk (e.g., *another desk*), inventing a nickname for him and cleverly working it—in a disguised form—into a seminar on one's research, etc. In short, I have seen a lot. Therefore, despite the fact that my acknowledgment section is unusually large, I cannot possibly name all the people who have contributed to making my experience at MIT an enjoyable and entertaining one.

First and foremost, I would like to thank my thesis supervisor Alan Willsky, who was the ideal advisor: not only a great scientist, but also a very enthusiastic and supportive mentor, providing broad guidance and posing interesting and important problems, yet allowing me virtually unrestricted freedom. His encouragement when research progressed well and his patience when it did not, were extremely helpful. Equally important was the fact that Alan was a fantastic collaborator: he was always up to date on every minor technical detail (in spite of his large number of students), which is why I never felt isolated in my research. His many suggestions on how to deal with difficult technical issues, as well as his vast knowledge of the literature, were invaluable. The speed with which he read drafts of papers and thesis chapters was

---

[1] Despite this fact, I probably hold another dubious record—namely, having the shortest dissertation. I think there have been people in the group whose theses had sentences longer than this whole document.

sometimes frightening; by following his comments and suggestions, one would typically improve the quality of a paper by orders of magnitude. His help and advice—and, I am sure, his reputation—were also instrumental to the success of my job-hunting campaign.

My co-supervisor Hamid Krim was equally supportive; I am very thankful to him for having an "open door" and for the endless hours we spent discussing research, politics, soccer, religion, and life in general. I am grateful to Hamid for inviting me to spend several days in North Carolina, and for his immense help during my job search.

The suggestions and advice of the two remaining committee members—Olivier Faugeras and Sanjoy Mitter—also greatly contributed both to my thesis and to finding a good academic position. I am indebted to Olivier for inviting me to visit his group at INRIA. I thoroughly enjoyed the month I spent in Antibes and Sophia-Antipolis, for which I would like to thank the whole ROBOTVIS group: Didier Bondyfalat, Sylvain Bougnoux, Rachid (Nour-Eddine) Deriche, Cyrille Gauclin, José Gomez, Pierre Kornprobst, Marie-Cécile Lafont, Diane Lingrand, Théo Papadopoulo, Nikos Paragios, Luc Robert, Robert Stahr, Thierry Viéville, and Imad Zoghlami.

My special thanks go to Stéphane Mallat for inviting me to give a seminar at École Polytechnique, and for his help in my job search. He has greatly influenced my professional development, both through a class on wavelets which he taught very engagingly and enthusiastically at MIT in 1994, and through his research, some of which was the basis for my Master's thesis.

I would like to thank Stuart Geman for suggesting the dynamic programming solution of Chapter 4, as well as for a number of other interesting and useful insights.

I am thankful for the stimulating discussions with Michele Basseville, Charlie Bouman, Yoram Bresler, Patrick Combettes, David Donoho, Al Hero, Mohamed Khodja, Jitendra Malik, Jean-Michel Morel, David Mumford, Igor Nikiforov, Pietro Perona, Jean-Christophe Pesquet, Guillermo Sapiro, Eero Simoncelli, Gil Strang, Allen Tannenbaum, Vadim Utkin, and Song-Chun Zhu, all of whom have contributed to improving the quality of my thesis and to broadening my research horizon.

In addition to the people I acknowledged above, Eric Miller, Peyman Milanfar, and Andy Singer generously shared with me many intricacies of the academic job search.

I thank the National Science Foundation and Alan Willsky for providing the financial support, through a Fellowship and a Research Assistantship, respectively. I thank Clem Karl, Paul Fieguth and Andrew Kim for all their help with computers; Ben Halpern for teaching me repeatedly and patiently how to use the fax machine, and for helping me to tame our color printer; Mike Daniel and Austin Frakt for sharing the LaTeXstyle file which was used to format this thesis; John Fisher for organizing a very informative reading group on learning theory; Asuman Koksal for making our office a little cozier; Andy Tsai for being a great officemate; Dewey Tucker for his help in generating my PowerPoint job talk; Jun Zhang and Tony Yezzi both for technical interactions and for in-depth discussions of the opera; Mike Schneider for sharing his encyclopedic knowledge (and peculiar tastes) of the art in general, and for educating me on a number of other topics, ranging from estimation and linear programming to botany and ethnography, as well as for volunteering to be my personal guide and chauffeur in Phoenix, AZ; Ron Dror for our numerous discussions of biological and computer vision; and Martin Wainwright for conveying some interesting facts about probability and natural image statistics, and for having even more peculiar tastes in arts than Mike. I also enjoyed the sometimes heated arguments with Khalid Daoudi over soccer, but I still think he was wrong about the penalty in that Brazil vs. Norway game! The unfortunate lack of a resident ballet expert in SSG was rectified by my summer employment at Alphatech, Inc. (in Burlington, MA), where I had a chance to exchange opinions and ballet tapes with Bob Tenney.

I enjoyed interacting with all other SSG group members and visitors, past and present. The ones whose names are not mentioned above are Lori Belcastro, François Bergeaud, Mickey Bhatia, Terrence Ho, Taylore Kelly, Junmo Kim, Alex Ihler, Bill Irving, Seema Jaggi, Rachel Learned, Cedric Logan, Mark Luettgen, Mike Perrott, Paula Place, John Richards, and Gerald Spica.

I thank my brother Nick for his leadership by example in showing me that an excellent PhD thesis *can* be finished even under very unfavorable circumstances, such as having two small children and a full-time job.

I am most grateful to my parents who gladly endured many sacrifices that accompanied our emigration, so that my brother and I could have a better life in the United States. Without their strength and courage I would not have had a chance to study in the best university in the world; without their tremendous support, I would not have been able to complete my studies. As I know how important my getting this degree was to my mother, I dedicate this thesis to her memory.

Contents

List of Figures

Chapter 1

Introduction

IN this chapter, we introduce the problem of segmentation and change detection addressed in this thesis, and describe the organization of the thesis.

1.1 Problem Description and Motivation

To segment a 1-D or 2-D signal means, roughly speaking, to partition the domain of its definition into several regions in such a way that the signal is homogeneous within each region and changes abruptly between regions. The exact meaning of the word "homogeneous" depends on the application: most often it means smoothly varying or constant intensity (this case is addressed in Chapters 3 and 4), or uniform texture (Chapters 4 and 5).

The objective of this thesis is to develop and analyze robust and fast image segmentation algorithms. They must be robust to pervasive, large-amplitude noise, which cannot be well characterized in terms of probabilistic distributions. This is because the applications of interest are exemplified by synthetic aperture radar (SAR) segmentation in which speckle noise is a well-known problem that has defeated many algorithms. (A prototypical SAR log-magnitude image of two textural regions–forest and grass–is shown in Figure 1.1.) The methods must also be robust to blur, because many imaging techniques result in smoothed images. For example, SAR image formation has a natural blur associated with it, due to the finite aperture used in forming the image.

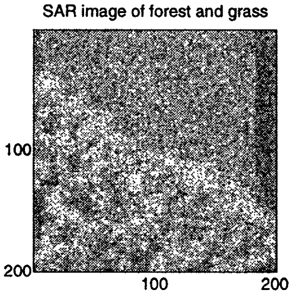


Figure 1.1. SAR image of trees and grass.

Image segmentation is closely related to restoration, that is, the problem of estimating an image based on its degraded observation. Indeed, the solution to one of these problems makes the other simpler: estimation is easier if the boundaries of homogeneous image regions are known, and vice versa, segmentation is easier once a good estimate of the image has been computed. It is therefore natural that many segmentation algorithms are related to restoration techniques, and in fact some methods combine the two, producing estimates of both the edge locations and image intensity [36, 44], as we will see in Chapter 2.

In describing any restoration or segmentation technique, the notion of *scale* is very important. Any such technique incorporates a scale parameter—either directly in the computation procedure, or implicitly as a part of the image model—which controls the smoothness of the estimate and/or sizes of the segmented regions. The precise definitions of scale, in several contexts, are given in Chapters 2 and 3; intuitively, changing the parameter from zero to infinity will produce a so-called *scale space*, i.e. a set of increasingly coarse versions of the input image. There are two approaches to generating a scale space: one starts with a probabilistic model, the other starts with a set of "common-sense" heuristics. The difference between the two is conceptual: they both may lead to the same algorithm [39, 44], producing the same scale space. In the former case, one would build a probabilistic model of images of interest [6, 32, 39] and proceed to derive an algorithm for computing the solution which is, in some probabilistic sense, optimal. For example, one could model images as piecewise constant functions with additive white Gaussian noise, and the edges (i.e. the boundaries separating the constant pieces of the function) as continuous curves whose total length is a random variable with a known distribution. Assigning larger probabilities to the occurrence of edges would correspond to larger scales in such a model, which will be illustrated in Chapter 4. Given a realization of this random field, the objective could be to compute the maximum likelihood estimates [66] of the edge locations. The main shortcoming of this approach is that a good model is unavailable in many applications, and that usually any realistic model yields a complicated objective functional to be optimized. Obtaining the optimal solution is therefore not computationally feasible, and one typically settles for a local maximum [6, 63]. An alternative to such probabilistic methods of generating scale spaces is to devise an algorithm using a heuristic description of images of interest. Stabilized Inverse Diffusion Equations (SIDEs), which are the main topic of this thesis, belong to this latter category.

SIDEs are motivated by the great recent interest in using evolutions specified by partial differential equations (PDE's) as image processing procedures for tasks such as restoration and segmentation, among others [1, 12, 35, 46, 49, 50, 55–57, 72]. The basic paradigm behind SIDEs, borrowed from [1, 35, 49, 72], is to treat the input image as the initial data for a diffusion-like differential equation. The unknown in this equation is usually a function of three variables: two spatial variables (one for each image dimension) and the scale—which is also called *time* because of the similarity of such equations to evolution equations encountered in physics. In fact, one of the starting points of this line of investigation was the observation [72] that smoothing an image with Gaussians of varying width is equivalent to solving the linear heat diffusion equation with the image as the initial condition. Specifically, the solution to the heat equation at time t is the convolution of its initial condition with a Gaussian of variance $2t$. Gaussian filtering has been used both to remove noise and as a pre-processor for edge detection procedures [9]. It has serious drawbacks, however: it displaces and removes important image features, such as edges, corners, and T-junctions. (An example of this behavior will be given in Chapter 2 (Figure 2.1).) The interpretation of Gaussian filtering as a linear diffusion led to the design of other, nonlinear, evolution equations, which better preserve these features [1,46,49,55–57]. For example, one motivation for the work of Perona and Malik in [49,50] is achieving both noise removal and edge enhancement through the use of an equation which in essence acts as an unstable inverse diffusion near edges and as a stable linear-heat-equation-like diffusion in homogeneous regions without edges.

The point of departure for the development of our SIDEs are the anisotropic diffusions introduced by Perona and Malik and described in the next chapter. In a sense that we will make both precise and conceptually clear, the evolutions that we introduce may be viewed as a conceptually limiting case of the Perona-Malik diffusions. These evolutions have discontinuous right-hand sides and act as inverse diffusions "almost everywhere" with stabilization resulting from the presence of the discontinuities in the right-hand side. As we will see, the scale space of such an equation is a family of segmentations of the original image, with larger values of the scale parameter t corresponding to segmentations at coarser resolutions.

Since "segmentation" may have different meanings in different contexts, we close this section by further clarifying which segmentation problems are addressed in this thesis and which are not. This also gives us an opportunity to mention four important application areas.

Automatic Target Recognition. Segmentation problems arise in many aspects of image processing for automatic target recognition. Some examples are the identification of tree-grass boundaries in synthetic aperture radar images [18] (an example given in the beginning of this section will be further treated in Chapter 3), the localization of hot targets in infrared radar [15], and the separation of foreground and background in laser radar (see [23] and references therein). Due to high rates of data acquisition, it is desirable for some of these problems that the algorithm be fast, in addition to being robust. As shown in Chapters 3 and 4, the algorithms presented in this thesis outperform existing segmentation algorithms in speed and/or robustness.

Segmentation of medical images. Given an image or a set of images resulting from a medical imaging procedure—such as ultrasound [13], magnetic resonance imaging [2,64], tomography [6,27], dermatoscopy [19,60]—it is necessary to extract certain objects of interest, e. g. an internal organ, a tumor, or a boundary between the gray matter and white matter. The main challenges of the segmentation problem depend on the object and on the imaging modality. For example, ultrasound imaging introduces both significant blurring and speckle noise [8, 22, 70], and so the corresponding segmentation algorithms must be robust to such degradations. Robustness of the algorithm introduced in this thesis is experimentally demonstrated in Chapters 3 and 4; Chapter 4 also contains its theoretical analysis.

Detection of abrupt changes in 1-D signals [3]. Application areas include analysis of electrocardiograms and seismic signals, vibration monitoring in mechanical structures, and quality control. Several synthetic 1-D examples are considered in Chapters 3 and 4.

Computer vision. In the computer vision literature, the term "segmentation" often refers to finding the contours of objects in natural images—i.e., photographic pictures of scenes which we are likely to see in our everyday life [16]. This is an important problem in low-level vision, because it has been universally accepted since [71] that segmenting the perceived scene plays an important role in human vision. However, noise and other types of degradation are usually not as significant here as in the medical and radar images; the main challenge is the variety of objects, shapes, and textures in a typical picture. This problem is therefore not directly addressed in the present thesis; applying the algorithms developed here to this problem is a topic for future research.

■ 1.2 Summary of Contributions and Thesis Organization

Chapter 2 is a review of several methods of image restoration and segmentation, each of which has important connections to the main contribution of this thesis, namely Stabilized Inverse Diffusion Equations (SIDEs). We mentioned in the previous section that SIDE evolutions produce nonlinear scale spaces, similarly to the equations in [1,49]. What sets SIDEs apart from other frameworks is the form of their right-hand sides which results in improved performance. This is shown in Chapter 3, both through experiments confirming the SIDEs' speed and robustness, and through proving a number of their useful properties, such as stability with respect to small changes in the input data. It is also explained how the SIDEs are related to several existing methods, such as Perona-Malik diffusions [49], shock filters [46], total variation minimization [6, 46, 58], the Geman-Reynolds functional [21], and region merging [43].

The main drawback of the methods which are based on heuristic descriptions of images of interest (rather than on probabilistic models) is that it is usually unclear how they perform on noisy images; experimentation is the most popular way of demonstrating robustness [5, 49, 56, 63]. This has been the case with diffusion-based methods: they are complicated nonlinear procedures, and therefore theoretical analysis of what happens when a random field is subjected to such a procedure is usually impossible. Unlike their predecessors, SIDEs turn out to be amenable to some probabilistic analysis, which is carried out in Chapter 4. It is shown that a specific SIDE finds in $N \log N$ time the maximum likelihood solutions to certain binary classification problems. The likelihood function for one of these problems is essentially a 1-D version of the Mumford-Shah functional [44]. Thus, an interesting link is established between diffusion equations, Mumford and Shah's variational formulation, and probabilistic models. The robustness of the SIDE is explained by showing that, in a certain special case, it is optimal with respect to an H_∞-like criterion—which, roughly speaking, means that the SIDE achieves the minimum worst-case error. The performance is also analyzed by computing bounds on the probabilities of errors in edge location estimates. To summarize, the main contribution of Chapter 4 is establishing a connection between diffusion-based methods and maximum likelihood edge detection, as well as extensive performance analysis.

Chapter 5 extends SIDEs to vector-valued images and images taking values on a circle. We argue that most of the properties derived in Chapter 3 carry over. These results are applicable to color segmentation, where the image value at every pixel is a three-vector of red, green, and blue values. We also apply our algorithm to texture segmentation, in which the vector image to be processed is formed by extracting features from the raw texture image, as well as to segmenting orientation images.

Possible directions of future research are proposed in Chapter 6.

Chapter 2

Preliminaries

A NUMBER of existing algorithms for image restoration and segmentation, which are related to SIDEs, are summarized in this chapter.

■ 2.1 Notation.

In this section, we describe the notation which is used in the current chapter. Most of this notation will carry over to the rest of the thesis; however, the large quantity of symbols needed will force us to adopt a slightly different notation in Chapter 4—which we will describe explicitly in Section 4.2.

We begin with the one-dimensional (1-D) case. The 1-D signal to be processed is denoted by $u^0(x)$. The superscript 0 is a reminder of the fact that the signal is to be processed via a partial differential equation (PDE) of the following form:

$$\begin{aligned} u_t &= \mathcal{A}_1(u, u_x, u_{xx}) \\ u(0,x) &= u^0(x). \end{aligned} \tag{2.1}$$

The variable t is called *scale* or *time*, and the solution $u(t,x)$ to (2.1), for $0 \le t < \infty$, is called *a scale space.* The partial derivatives with respect to t and x are denoted by subscripts, and $\mathcal{A}_1$ is an operator. The scale space is called *linear* (*nonlinear*) if $\mathcal{A}_1$ is a linear (nonlinear) operator.

Similarly, an image $u^0(x,y)$ depending on two spatial variables, x and y, will be processed using a PDE of the form

$$\begin{aligned} u_t &= \mathcal{A}_2(u, u_x, u_y, u_{xx}, u_{yy}, u_{xy}) \\ u(0,x,y) &= u^0(x,y), \end{aligned} \tag{2.2}$$

which generates the scale space $u(t,x,y)$, for $0 \le t < \infty$. In the PDEs we consider, the right-hand side will sometimes involve the *gradient* and *divergence* operators. The gradient of $u(t,x,y)$ is the two-vector consisting of the partial derivatives of u with respect to the spatial variables x and y:

$$\nabla u \overset{\text{def}}{=} (u_x, u_y)^T, \tag{2.3}$$

where the superscript T denotes the transpose of a vector. The norm of the gradient is:

$$|\nabla u| \overset{\text{def}}{=} \sqrt{u_x^2 + u_y^2} \tag{2.4}$$

The divergence of a vector function $(u(x,y), v(x,y))^T$ is:

$$\vec{\nabla} \cdot \begin{pmatrix} u \\ v \end{pmatrix} \overset{\text{def}}{=} u_x + v_y. \tag{2.5}$$

We also consider semi-discrete versions of (2.1) and (2.2), obtained by discretizing the spatial variables and leaving t continuous. Specifically, an N-point 1-D discrete signal to be processed is denoted by $\mathbf{u}^0$; it is an element of the N-dimensional vector space $I\!R^N$. We exclusively reserve **boldface** letters for vectors—i.e., discrete signals and images. The vector $\mathbf{u}^0$ is the initial condition to the following N-dimensional ordinary differential equation (ODE):

$$\begin{aligned} \dot{\mathbf{u}}(t) &= \mathcal{B}_1(\mathbf{u}(t)) \\ \mathbf{u}(0) &= \mathbf{u}^0, \end{aligned} \tag{2.6}$$

where $\mathbf{u}(t)$ is the corresponding scale space, and $\dot{\mathbf{u}}(t)$ is its derivative with respect to t. We denote the entries of an N-point signal by the same symbol as the signal itself, with additional subscripts 1 through N:

$$\begin{aligned} \mathbf{u}^0 &= (u_1^0, u_2^0, \ldots, u_{N-1}^0, u_N^0)^T; \\ \mathbf{u}(t) &= (u_1(t), u_2(t), \ldots, u_{N-1}(t), u_N(t))^T. \end{aligned}$$

Since most operators $\mathcal{B}_1$ of interest will involve first differences of the form $u_{n+1} - u_n$, it will simplify our notation to also define non-existent samples u_0 and u_{N+1}. Thus, all vectors will implicitly be $(N+2)$-dimensional. Typically, we will take $u_0 = u_1$ and $u_{N+1} = u_N$. We emphasize that subscripts 0 through $N+1$ will always denote the samples of a signal, whereas the superscript 0 will be reserved exclusively to denote the signal which is the initial condition of a differential equation.

We similarly denote an N-by-N image to be processed by $\mathbf{u}^0 \in I\!R^{N^2}$; it will always be clear from the context whether $\mathbf{u}^0$ refers to a 1-D or a 2-D discrete signal. The corresponding system of ODEs is

$$\begin{aligned} \dot{\mathbf{u}}(t) &= \mathcal{B}_2(\mathbf{u}(t)) \\ \mathbf{u}(0) &= \mathbf{u}^0, \end{aligned} \tag{2.7}$$

where $\mathbf{u}^0$ and $\mathbf{u}(t)$ are matrices whose entries in the i-th row and j-th column are $u_{i,j}^0$ and $u_{i,j}(t)$, respectively.

The operators $\mathcal{B}_1$ and $\mathcal{B}_2$ will typically be the negative gradient of some energy functional, which we will denote by $\mathcal{E}(\mathbf{u})$. This energy will depend on the first differences of $\mathbf{u}$ in the following way:

$$\mathcal{E}(\mathbf{u}) = \sum_{(s,r)\in\mathcal{N}} E(u_s - u_r), \tag{2.8}$$

where

- E is an even function;
- s and r are single indices if $\mathbf{u}$ is a 1-D signal and pairs of indices if $\mathbf{u}$ is a 2-D image;
- $\mathcal{N}$ is the list of all neighboring pairs of pixels: s and r are neighbors if and only if $(s,r) \in \mathcal{N}$.

We will use the following neighborhood structure in 1-D:

$$\mathcal{N} = \{(n, n+1)\}_{n=1}^{N-1}. \tag{2.9}$$

In other words, the sample at n has two neighbors: at $n-1$ and at $n+1$. We use a similar neighborhood structure in 2-D, where each pixel (i,j) has four neighbors: $(i-1,j)$, $(i+1,j)$, $(i,j-1)$, and $(i,j+1)$.

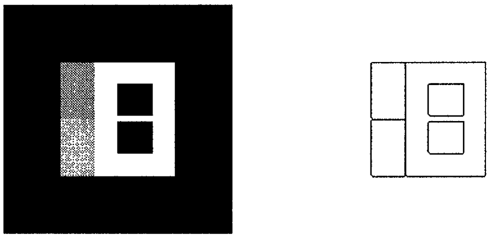
2.2 Linear and Non-linear Diffusions.

To understand the conceptual basis of SIDEs, it is useful to briefly review one of the lines of thought that has spurred work in evolution-based methods for image analysis. In [72] Witkin proposed filtering an original image $u^0(x,y)$ with Gaussian kernels of variance $2t$, to result in a one-parameter family of images $u(t,x,y)$ he referred to as "a scale space". This filtering technique has both a very important interpretation and a number of significant limitations that inspired the search for alternative scale spaces, better adapted to edge detection and image segmentation. In particular, a major limitation is that linear Gaussian smoothing blurs and displaces edges, merges boundaries of objects that are close to each other, and removes edge junctions [55], as illustrated in Figure 2.1.

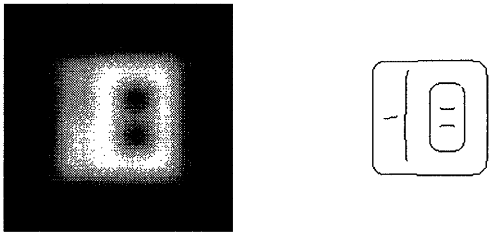
However, the important insight found, for example, in [35], is that the family of images $u(t,x,y)$ is the solution of the linear heat equation with $u^0(x,y)$ as the initial data:

$$\begin{aligned} u_t &= u_{xx} + u_{yy} \\ u(0,x,y) &= u^0(x,y), \end{aligned} \tag{2.10}$$

where the subscripts denote partial derivatives. This insight led to the pursuit and development of a new paradigm for processing images via the evolution of nonlinear



(a) An image. (b) Its edges.



(c) Blurred image. (d) Corresponding edges.

Figure 2.1. (a) An artificial image; (b) the edges corresponding to the image in (a); (c) the image in (a) blurred with a Gaussian kernel; (d) the edges corresponding to the blurred image. Note that T-junctions are removed, corners are rounded, and two black squares are merged together. The edges here are the maxima of the absolute value of the gradient.

PDEs [1,46,49,50,56] which effectively lift the limitations of the linear heat equation. For example, in [49,50], Perona and Malik propose to achieve both noise removal and edge enhancement through the use of a non-uniform diffusion which in essence acts as an unstable inverse diffusion near edges and as a stable linear-heat-equation-like diffusion in homogeneous regions without edges:

$$\begin{aligned} u_t &= \vec{\nabla} \cdot \{G(|\nabla u|)\nabla u\}, \\ u(0,x,y) &= u^0(x,y), \end{aligned} \tag{2.11}$$

where $\vec{\nabla}$ and ∇ are the divergence (2.5) and gradient (2.3), respectively. The nonlinear

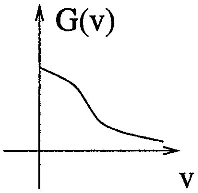


Figure 2.2. The G function from the right-hand side of the Perona-Malik equation (2.11).

diffusion coefficient $G(|\nabla u|)$ is chosen so as to suppress diffusion in the regions of high gradient which are identified with edges, and to encourage diffusion in low-gradient regions identified with noise (Figure 2.2). More formally, G is a nonnegative monotonically decreasing function with $G(0) = 1$. (Note that if G were identically equal to 1, then (2.11) would turn into the linear heat equation (2.10), since $\vec{\nabla} \cdot (\nabla u) = u_{xx} + u_{yy}$.)

To simplify the analysis of the behavior of this equation near edges, we re-write it below in one spatial dimension; however, the statements we make also apply to 2-D.

$$\begin{aligned} u_t &= \frac{\partial}{\partial x}\{F(u_x)\} \\ u(0,x) &= u^0(x), \end{aligned} \tag{2.12}$$

where $F(u_x) = G(|u_x|)u_x$, i.e., F is odd, and tends to zero at infinity. Perona and Malik also impose that F have a unique maximum at some location K (Figure 2.3). This constant K is the threshold between diffusion and enhancement, in the following sense. If, for a particular time $t = t_0$, we define an "edge" of $u(t_0, x)$ as an inflection point with the property $u_x u_{xxx} < 0$, then a simple calculation shows that all such edges where $|u_x| < K$ will be diminished by (2.12)—i.e. $|u_x|$ will be reduced, while the larger edges, with $|u_x| > K$, will be enhanced. It has been observed [34] that the numerical implementations of (2.12) do not exactly exhibit this behavior, although they do produce temporary enhancement of edges, resulting in both noise removal and scale spaces in which the edges are much more stable across scale than in linear scale spaces. As Weickert pointed out in [69], "a scale-space representation cannot perform better than its discrete realization". These observations naturally led to a closer analysis (described in the next chapter) of a semi-discrete counterpart of (2.12), i.e., of the following system of ordinary differential equations:

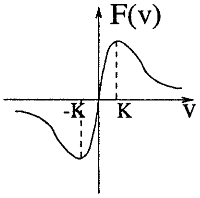


Figure 2.3. The F function from the right-hand side of the Perona-Malik equation (2.12).

$$\begin{aligned} \dot{u}_n &= F(u_{n+1} - u_n) - F(u_n - u_{n-1}), \; n = 1, \ldots, N, \\ \mathbf{u}(0) &= \mathbf{u}^0, \end{aligned} \tag{2.13}$$

where $\mathbf{u}^0 = (u_1^0, \ldots, u_N^0)^T \in I\!R^N$ is the signal to be processed, and where the conventions $u_{N+1} = u_N$ and $u_0 = u_1$ are used.

The 2-D semi-discrete version of the Perona-Malik equation is similar:

$$\begin{aligned} \dot{u}_{ij} &= F(u_{i+1,j} - u_{ij}) - F(u_{ij} - u_{i-1,j}) \\ &+ F(u_{i,j+1} - u_{ij}) - F(u_{ij} - u_{i,j-1}), \\ \mathbf{u}(0) &= \mathbf{u}^0, \end{aligned}$$

with $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, N$, and with the conventions $u_{0,j} = u_{1,j}$, $u_{N+1,j} = u_{N,j}$, $u_{i,0} = u_{i,1}$ and $u_{i,N+1} = u_{i,N}$.

■ 2.3 Region Merging Segmentation Algorithms.

It is shown in the next chapter that SIDEs can be classified under a very broad category of multiscale region merging segmentation methods.

The goal of image segmentation is to partition the domain Ω of definition of a given image $\mathbf{u}^0$ into several disjoint regions $O_1, \ldots, O_k$ ($\cup_{i=1}^k O_i = \Omega$), such that $\mathbf{u}^0$ is, in some sense, homogeneous within each region. Many segmentation algorithms also provide a filtered version $\mathbf{u}$ of $\mathbf{u}^0$, which is in essence a smoothing of $\mathbf{u}^0$ within each region and has abrupt changes between regions. A boundary between two such neighboring regions is called *an edge*. If a segmentation E_2 can be obtained from a segmentation E_1 by erasing edges, it is said that E_2 is *coarser* than E_1. A *multiscale segmentation algorithm* provides a hierarchy of segmentations which get progressively coarser. Morel and Solimini's [43] definition of a generic multiscale region merging algorithm for image segmentation is paraphrased and augmented below.

1. Initialize the algorithm with the finest possible segmentation (i.e., each pixel is a separate region). Fix the scale t at a very small value t_0.
2. Merge all pairs of regions whose merging "improves" the segmentation. If the coarsest segmentation is obtained (i.e., the whole domain Ω is the one and only region), stop.
3. If necessary, update the estimate $\mathbf{u}$.
4. Increment the scale parameter t.
5. Go to Step 2.

Numerous algorithms have been constructed that fall under this, very general, paradigm. Their principal points of difference are Step 3 and the merging criterion used in Step 2. Two instances of such algorithms are Pavlidis' algorithm of 1972 [47] and Koepfler, Lopez, and Morel's algorithm of 1994 [36].

Example 2.1. *Pavlidis' algorithm.*

Step 2 of Pavlidis' algorithm consists of merging two neighboring regions, O_i and O_j, if the variance of $\mathbf{u}^0$ over $O_i \cup O_j$ is less than t. Step 3 is omitted. ■

Example 2.2. *The algorithm of Koepfler, Lopez, and Morel.*

Koepfler, Lopez, and Morel use the piecewise-constant Mumford-Shah model [44] as their merging criterion. They form the filtered image $\mathbf{u}$ by assigning to every pixel in a region O_i the average value of $\mathbf{u}^0$ over that region. Two regions, O_i and O_j, with respective average values u_i and u_j, are merged by removing the boundary between them and replacing both u_i and u_j with their weighted average, $(|O_i|u_i + |O_j|u_j)/(|O_i| + |O_j|)$, provided that the global energy is reduced:

$$(\mathbf{u} - \mathbf{u}^0)^T(\mathbf{u} - \mathbf{u}^0) + tl.$$

Here, $|O_p|$ is the number of pixels in the region O_p and l is the total length of all the edges.

This method admits fast numerical implementations and has been experimentally shown to be robust to white Gaussian noise. However, as we will illustrate, the quadratic penalty on the disagreement between the estimate $\mathbf{u}$ and the initial data $\mathbf{u}^0$ renders it ineffective against more severe noise, such as speckle encountered in SAR images. ■

Note that region merging methods do not allow edges to be created. Thus, decisions made in the beginning of an algorithm cannot be undone later. A slight modification of such methods results in split-and-merge methods, which combine region growing with region splitting [43].

■ 2.4 Shock Filters and Total Variation.

We will see in the next chapter that SIDEs blend together elements of the total variation minimization [6, 56, 58] and shock filters [46]. A shock filter is a PDE which was introduced by Osher and Rudin in [46] to achieve signal and image enhancement (de-blurring). Since its 2-D version will not be needed in this thesis, we restrict our discussion to one spatial dimension:

$$\begin{aligned} u_t &= -|u_x| f(u_{xx}), \\ u(0,x) &= u^0(x), \end{aligned} \tag{2.14}$$

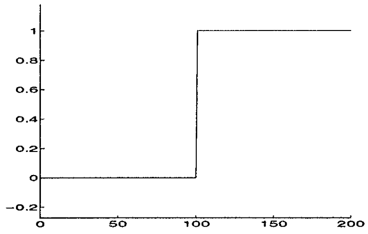
where $f(0) = 0$, and the function f has the same sign as its argument: $\mathrm{sgn}(f(u_{xx})) = \mathrm{sgn}(u_{xx})$ for $u_{xx} \neq 0$. This PDE has the property of developing shocks (jumps) near the points of inflection of the initial condition. The most serious limitation of the shock filters, acknowledged in [46], is non-robustness to noise. In fact, they enhance noise, together with the "useful" edges. There are, however, stable numerical schemes, which—similarly to the semi-discrete Perona-Malik equation—do not exhibit certain properties of the underlying PDE. In particular, for certain choices of f, the scheme presented in [46] sharpens noise-free signals, but leaves noisy signals virtually unchanged. This is because every local extremum remains stationary under this scheme. This behavior is illustrated in Figure 2.4, which presents the results of the simulations for the equation

$$u_t = -|u_x| \,\mathrm{sgn}(u_{xx}). \tag{2.15}$$

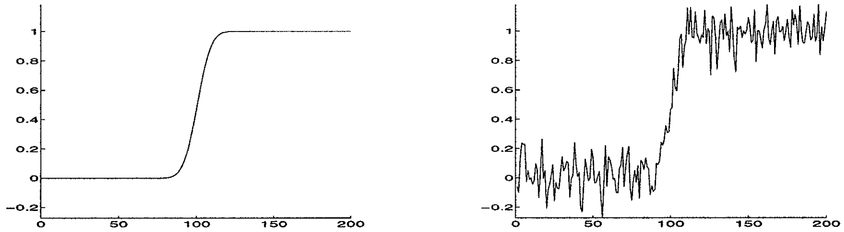
For other choices of f, the numerical scheme of [46], in addition to keeping the local extrema stationary, creates other local extrema and is unstable, which is illustrated in Figure 2.5 for the equation

$$u_t = -|u_x| u_{xx}. \tag{2.16}$$

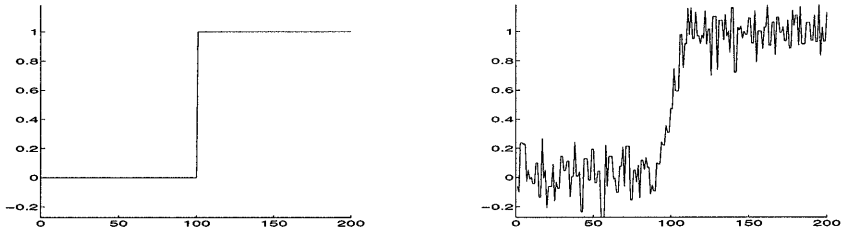
Total variation minimization is another restoration technique, developed by the same authors in [56] and independently by Bouman and Sauer in [6, 58] in a different



(a) A unit step signal.



(b) A blurred step. (c) The blurred step with additive noise.



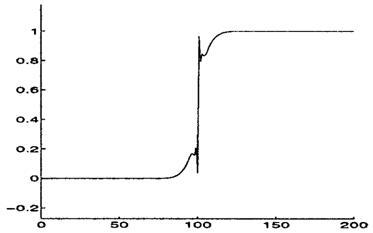
(d) Shock filtering of (b). (e) Shock filtering of (c).

Figure 2.4. (b) Gaussian blurring of the signal depicted in (a). (c) Signal depicted in (b), with additive white Gaussian noise of variance 0.1. (d) The steady state of the shock filter (2.15), with the signal (b) as the initial condition. The reconstruction is perfect, modulo numerical errors. (e) The steady state of the shock filter (2.15), with the signal (c) as the initial condition. It is virtually the same as (c), since all extrema remain stationary.

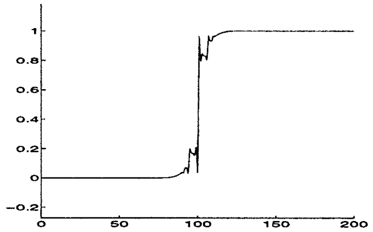
context and in a somewhat different form. We start with Bouman and Sauer's work, since it was chronologically first, and since—as we will see in the next chapter—it is conceptually closer to the results presented in this thesis.

The objective of [6, 58] is reconstructing an image $\mathbf{u}$ from its tomographic projections $\mathbf{u}^0$. The authors consider transmission tomography, where the projection data are in the form of the number of photons detected after passing through an absorptive material. In other words, $\mathbf{u}^0$ is the number of photon counts for each angle and displacement. Bouman and Sauer use a probabilistic setting, where the photon counts are Poisson random variables, independent among angles and displacements. They derive an expression for the log likelihood function $L(\mathbf{u}|\mathbf{u}^0)$, and seek the maximum *a posteriori* [66] estimate $\hat{\mathbf{u}}$ of $\mathbf{u}$:

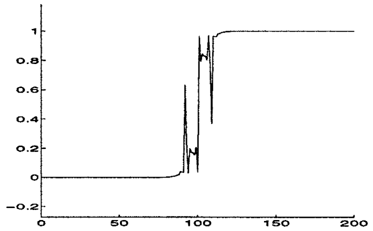
$$\hat{\mathbf{u}} = \arg\max_{\mathbf{u}}\{L(\mathbf{u}|\mathbf{u}^0) + \mathcal{E}(\mathbf{u})\}, \tag{2.17}$$



(a) Five iterations of the shock filter (2.16).



(a) Ten iterations of the shock filter (2.16).



(a) Eighteen iterations of the shock filter (2.16).

Figure 2.5. Filtering the blurred unit step signal of Figure 2.4, (b) with the shock filter (2.16): (a) 5 iterations, (b) 10 iterations, (c) 18 iterations. Spurious maxima and minima are created; the unit step is never restored.

where $\mathcal{E}(\mathbf{u})$ is the logarithm of the prior density function of $\mathbf{u}$. They propose the following prior model:

$$\mathcal{E}(\mathbf{u}) = \gamma \sum_{(s,r)\in\mathcal{N}} |u_s - u_r|, \tag{2.18}$$

where $\mathcal{N}$ is the list of all neighboring pairs of pixels, and γ is a constant.

The optimization problem (2.17) cannot be solved by gradient descent, since $\mathcal{E}(\mathbf{u})$ is not differentiable at the points where $u_s = u_r$ for $(s, r) \in \mathcal{N}$. Bouman and Sauer improve upon the existing iterative methods for solving non-differentiable optimization problems [4, 54, 73] by introducing a technique which they call *segmentation based optimization.* The basic idea is to combine a Gauss-Seidel type approach [59] with a split-and-merge segmentation strategy. Specifically, if the image values at two neighboring locations are both equal to some number α, then α is changed until a minimum of the objective function (2.17) is achieved. Similarly, if several neighboring pixels $u_{s_1} = u_{s_2} = \ldots = u_{s_i}$ have the same value, they are grouped together and changed together by moving along the hyperplane $\{\mathbf{u} : u_{s_1} = u_{s_2} = \ldots = u_{s_i}\}$ until a minimum of the objective function (2.17) is achieved. After each pixel of the image is visited in such a manner, a "split" iteration follows, where each pixel is freed to seek its own conditionally optimal value. This approach is theoretically justified and extended in Chapter 3, where it is shown that the steepest descent for a non-differentiable energy function such as (2.18) is a differential equation which automatically merges pixels, thereby segmenting the underlying image.

We also point out that the continuous version of the energy (2.18) is

$$\begin{aligned} &\int |u_x|\, dx && \text{in 1-D,} \\ \text{and } &\int\!\!\int |\nabla u|\, dx\, dy && \text{in 2-D,} \end{aligned}$$

and is called *the total variation of u*. Its constrained minimization was used in [56] for image restoration. The restored version $u(x, y)$ of an image $u^0(x, y)$ was computed by solving the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \int\!\!\int |\nabla u|\, dx\, dy \\ \text{subject to} \quad & \int\!\!\int (u - u^0)\, dx\, dy = 0 \\ \text{and} \quad & \int\!\!\int (u - u^0)^2\, dx\, dy = \sigma^2. \end{aligned} \tag{2.19}$$

■ 2.5 Constrained Restoration of Geman and Reynolds.

We will see in the next chapter that a 1-D SIDE is the gradient descent equation for the global energy $\mathcal{E}(\mathbf{u}) = \sum E(u_{i+1} - u_i)$, where $E(v)$ is concave everywhere except at zero and non-differentiable at zero, and looks like Υ (for example, $E(v) = arctan(v)$ or $E(v) = 1 - (1 + |v|)^{-1}$—see Figure 2.6). This energy is similar to the first term of the image restoration model of D. Geman and Reynolds [21]. It is also interesting to note that the potential function of the Gibbs distribution learned from natural images in [75] has the same basic Υ-shape. This indicates that the functionals involving such a term may be the right ones for modeling natural images.

■ 2.6 Conclusion.

An exhaustive survey of variational models in image processing is beyond the scope of this thesis. A much more complete bibliography can be found in [43]. In particular, Chapter 3 of [43] contains a very nice discussion of region merging segmentation algorithms, starting with Brice and Fennema's [7] and Pavlidis' [47], which may be considered as ancestors to both [36], snakes [31], and SIDEs. Examples of more recent algorithms, not covered in [43], are [16] and [24]. Another important survey text,

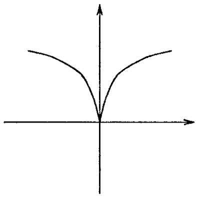


Figure 2.6. The SIDE energy function, also encountered in the models of Geman and Reynolds, and Zhu and Mumford.

which also contains a wealth of references both on variational methods and nonlinear diffusions, is [55].

Chapter 3

Image Segmentation with Stabilized Inverse Diffusion Equations

3.1 Introduction.

IN this chapter, we introduce the Stabilized Inverse Diffusion Equations (SIDEs), as well as illustrate their speed and robustness, in comparison with some of the methods reviewed in Chapter 2. As we mentioned in the previous chapter, the starting point for the development of SIDEs were image restoration and segmentation procedures based on PDEs of evolution [1,12,46,49,50,55–57,72]. We observed that the numerical schemes for solving such equations do not necessarily exhibit the behavior of the equations themselves. We therefore concentrate in this thesis on semi-discrete scale spaces (i.e., continuous in scale and discrete in space). More specifically, SIDEs, which are the main focus and contribution of this thesis, are a new family of semi-discrete evolution equations which stably sharpen edges and suppress noise. We will see that SIDEs may be viewed as a conceptually limiting case of Perona-Malik diffusions which were reviewed in the previous chapter. SIDEs have discontinuous right-hand sides and act as inverse diffusions "almost everywhere", with stabilization resulting from the presence of discontinuities in the vector field defined by the evolution. The scale space of such an equation is a family of segmentations of the original image, with larger values of the scale parameter t corresponding to segmentations at coarser scales. Moreover, in contrast to continuous evolutions, the ones introduced here naturally define a sequence of logical "stopping times", i.e. points along the evolution endowed with useful information, and corresponding to times at which the evolution hits a discontinuity surface of its defining vector field.

In the next section we begin by describing a convenient mechanical analog for the visualization of many spatially-discrete evolution equations, including discretized linear or nonlinear diffusions such as that of Perona and Malik, as well as the discontinuous equations that we introduce in Section 3.3. The implementation of such a discontinuous equation naturally results in a recursive region merging algorithm. Because of the discontinuous right-hand side of SIDEs, some care must be taken in defining solutions, but as we show in Section 3.4, once this is done, the resulting evolutions have a number of important properties. Moreover, as we have indicated, they lead to very effective algorithms for edge enhancement and segmentation, something that is demonstrated in Section 3.5. In particular, as we will see, they can produce sharp enhancement of edges in high noise as well as accurate segmentations of very noisy imagery such as SAR and ultrasound imagery subject to severe speckle. In Section 3.6, we point out its principal differences from Koepfler, Lopez, and Morel's [36] region merging procedure for minimizing the Mumford-Shah functional [44]. The rest of that section is devoted to exploring the links with other important work in the field reviewed in Chapter 2: the total variation approach [6,56,58]; shock filters of Osher and Rudin [46]; the robust variational formulation of D. Geman and Reynolds [21]; and the stochastic modeling approach of Zhu and Mumford [75].

■ 3.2 A Spring-Mass Model for Certain Evolution Equations.

As we indicated in the introduction, the focus of this chapter is on discrete-space, continuous-time evolutions of the following general form:

$$\begin{aligned} \dot{\mathbf{u}}(t) &= \mathcal{F}(\mathbf{u})(t), \\ \mathbf{u}(0) &= \mathbf{u}^0, \end{aligned} \tag{3.1}$$

where $\mathbf{u}$ is either a discrete sequence consisting of N samples ($\mathbf{u} = (u_1, \ldots, u_N)^T \in I\!R^N$), or an N-by-N image whose j-th entry in the i-th row is u_{ij} ($\mathbf{u} \in I\!R^{N^2}$). The initial condition $\mathbf{u}^0$ corresponds to the original signal or image to be processed, and $\mathbf{u}(t)$ then represents the evolution of this signal/image at time (scale) t, resulting in a scale-space family for $0 \leq t < \infty$.

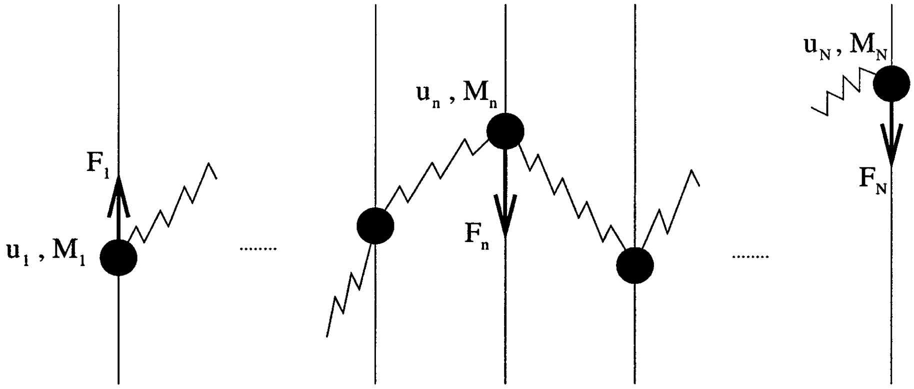


Figure 3.1. A spring-mass model.

The nonlinear operators $\mathcal{F}$ of interest in this chapter can be conveniently visualized through the following simple mechanical model. For the sake of simplicity in visualization, let us first suppose that $\mathbf{u} \in I\!R^N$ is a one-dimensional (1-D) sequence, and interpret $\mathbf{u}(t) = (u_1(t), \ldots, u_N(t))^T$ in (3.1) as the vector of vertical positions of the N particles of masses $M_1, \ldots, M_N$, depicted in Figure 3.1. The particles are forced to move along N vertical lines. Each particle is connected by springs to its two neighbors (except the first and last particles, which are only connected to one neighbor.) Every spring whose vertical extent is v has energy $E(v)$, i.e., the energy of the spring between the n-th and $(n+1)$-st particles is $E(u_{n+1}-u_n)$. We impose the usual requirements on this energy function:

$$\begin{aligned} E(v) &\geq 0, \\ E(0) &= 0, \\ E'(v) &\geq 0 \text{ for } v>0, \\ E(v) &= E(-v). \end{aligned} \tag{3.2}$$

Then the derivative of $E(v)$, which we refer to as "the force function" and denote by $F(v)$, satisfies

$$\begin{aligned} F(0) &= 0, \\ F(v) &\geq 0 \text{ for } v>0, \\ F(v) &= -F(-v). \end{aligned} \tag{3.3}$$

We also call $F(v)$ a "force function" and $E(v)$ an "energy" if $-E(v)$ satisfies (3.2) and $-F(v)$ satisfies (3.3). We make the movement of the particles non-conservative by stopping it after a small period of time Δt and re-starting with zero velocity. (Note that this will make our equation non-hyperbolic.) It is assumed that during one such step, the total force $F_n = -F(u_n-u_{n+1}) - F(u_n-u_{n-1})$, acting on the n-th particle, stays approximately constant. The displacement during one iteration is proportional to the product of acceleration and the square of the time interval:

$$u_n(t+\Delta t) - u_n(t) = \frac{(\Delta t)^2}{2}\frac{F_n}{M_n}.$$

Letting $\Delta t \to 0$, while fixing $\frac{2M_n}{\Delta t} = m_n$, where m_n is a positive constant, leads to

$$\dot{u}_n = \frac{1}{m_n}(F(u_{n+1}-u_n) - F(u_n-u_{n-1})), \quad n=1,2,\ldots,N, \tag{3.4}$$

with the conventions $u_0 = u_1$ and $u_{N+1} = u_N$ imposed by the absence of springs to the left of the first particle and to the right of the last particle. We will refer to m_n as "the mass of the n-th particle" in the remainder of the thesis. Note that Equation (3.4) is a (weighted) gradient descent equation for the following global energy:

$$\mathcal{E}(\mathbf{u}) = \sum_{i=1}^{N-1} E(u_{i+1}-u_i). \tag{3.5}$$

The examples below, where $m_n = 1$, clearly illustrate these notions.

Example 3.1. *Linear heat equation.*

A linear force function $F(v) = v$ leads to the semi-discrete linear heat equation

$$\dot{u}_n = u_{n+1} - 2u_n + u_{n-1}.$$

This corresponds to a simple discretization of the 1-D linear heat equation and results in evolutions which produce increasingly low-pass filtered and smoothed versions of the original signal $\mathbf{u}^0$. ■

In general, $F(v)$ is called a "diffusion force" if, in addition to (3.3), it is monotonically increasing:

$$v_1 < v_2 \quad \Rightarrow \quad F(v_1) < F(v_2), \tag{3.6}$$

which is illustrated in Figure 3.2(a). We shall call the corresponding energy a "diffusion energy" and the corresponding evolution (3.4) a "diffusion". The evolution in Example 3.1 is clearly a diffusion. We call $F(v)$ an "inverse diffusion force" if $-F(v)$ satisfies Equations (3.3) and (3.6), as illustrated in Figure 3.2(b). The corresponding evolution (3.4) is called an "inverse diffusion". Inverse diffusions have the characteristic of enhancing abrupt differences in $\mathbf{u}$ corresponding to "edges" in the 1-D sequence. Such pure inverse diffusions, however, lead to unstable evolutions (in the sense that they greatly amplify arbitrarily small noise). The following example, which is prototypical of the examples considered by Perona and Malik, defines a stable evolution that captures at least some of the edge enhancing characteristics of inverse diffusions.

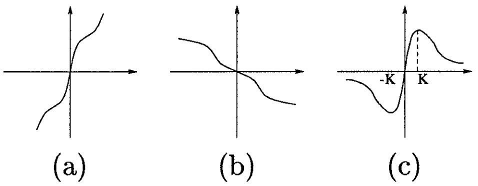


Figure 3.2. Force functions: (a) diffusion; (b) inverse diffusion; (c) Perona-Malik.

Example 3.2. *Perona-Malik equations.*

Taking $F(v) = v \exp(-(\frac{v}{K})^2)$, as illustrated in Figure 3.2(c), yields a 1-D semi-discrete (continuous in scale and discrete in space) version of the Perona-Malik equation (see equations (3.3), (3.4), and (3.12) in [50]). In general, given a positive constant K, a force $F(v)$ will be called "Perona-Malik force of thickness K" if, in addition to (3.3), it satisfies the following conditions:

$$\begin{gathered} F(v) \text{ has a unique maximum at } v = K, \\ F(v_1) = F(v_2) \quad \Rightarrow \quad (|v_1| - K)(|v_2| - K) < 0. \end{gathered} \tag{3.7}$$

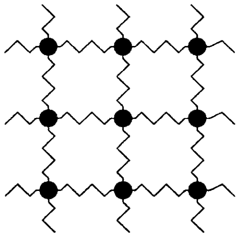


Figure 3.3. Spring-mass model in 2-D (view from above).

We shall call the corresponding energy a "Perona-Malik energy" and the corresponding evolution equation a "Perona-Malik equation of thickness K". As Perona and Malik demonstrate (and as can also be inferred from the results in the present thesis), evolutions with such a force function act like inverse diffusions in the regions of high gradient and like usual diffusions elsewhere. They are stable and capable of achieving some level of edge enhancement depending on the exact form of $F(v)$. ■

Finally, to extend the mechanical model of Figure 3.1 to images, we simply replace the sequence of vertical lines along which the particles move with an N-by-N square grid of such lines, as shown in Figure 3.3. The particle at location (i, j) is connected by springs to its four neighbors: $(i-1, j)$, $(i, j+1)$, $(i+1, j)$, $(i, j-1)$, except for the particles in the four corners of the square (which only have two neighbors each), and the rest of the particles on the boundary of the square (which have three neighbors). This arrangement is reminiscent of (and, in fact, was suggested by) the resistive network of Figure 8 in [49]. The analog of Equation (3.4) for images is then:

$$\begin{aligned} \dot{u}_{ij} &= \frac{1}{m_{ij}}(F(u_{i+1,j} - u_{ij}) - F(u_{ij} - u_{i-1,j}) \\ &+ F(u_{i,j+1} - u_{ij}) - F(u_{ij} - u_{i,j-1}), \end{aligned} \tag{3.8}$$

with $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, N$, and the conventions $u_{0,j} = u_{1,j}$, $u_{N+1,j} = u_{N,j}$, $u_{i,0} = u_{i,1}$ and $u_{i,N+1} = u_{i,N}$ imposed by the absence of springs outside of $1 \le i \le N$, $1 \le j \le N$.

■ 3.3 Stabilized Inverse Diffusion Equations (SIDEs): The Definition.

In this section, we introduce a discontinuous force function, resulting in a system (3.4) that has discontinuous right-hand side (RHS). Such equations received much attention in control theory because of the wide usage of relay switches in automatic control systems [17, 67]. More recently, deliberate introduction of discontinuities has been used in control applications to drive the state vector onto lower-dimensional surfaces in the state space [67]. As we will see, this objective of driving a trajectory onto a lower-dimensional surface also has value in image analysis and in particular in image segmentation. Segmenting a signal or image, represented as a high-dimensional vector

$\mathbf{u}$, consists of evolving it so that it is driven onto a comparatively low-dimensional subspace which corresponds to a segmentation of the signal or image domain into a small number of regions.

The type of force function of interest to us here is illustrated in Figure 3.4. More precisely, we wish to consider force functions $F(v)$ which, in addition to (3.3), satisfy the following conditions:

$$\begin{aligned} F'(v) &\leq 0 \quad \text{for} \quad v \neq 0, \\ F(0^+) &> 0 \\ F(v_1) = F(v_2) &\quad \Leftrightarrow \quad v_1 = v_2. \end{aligned} \tag{3.9}$$

Contrasting this form of a force function to the Perona-Malik function in Figure 3.2,

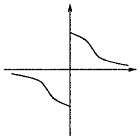


Figure 3.4. Force function for a stabilized inverse diffusion equation.

we see that in a sense one can view the discontinuous force function as a limiting form of the continuous force function in Figure 3.2(c), as $K \to 0$. However, because of the discontinuity at the origin of the force function in Figure 3.4, there is a question of how one defines solutions of Equation (3.4) for such a force function. Indeed, if Equation (3.4) evolves toward a point of discontinuity of its RHS, the value of the RHS of (3.4) apparently depends on the direction from which this point is approached (because $F(0^+) \neq F(0^-)$), making further evolution non-unique. We therefore need a special definition of how the trajectory of the evolution proceeds at these discontinuity points.[1] For this definition to be useful, the resulting evolution must satisfy well-posedness properties: the existence and uniqueness of solutions, as well as stability of solutions with respect to the initial data. In the rest of this section we describe how to define solutions to (3.4) for force functions (3.9). Assuming the resulting evolutions to be well-posed, we demonstrate that they have the desired qualitative properties, namely that they both are stable and also act as inverse diffusions and hence enhance edges. We address the issue of well-posedness and other properties in Section 3.4.

Consider the evolution (3.4) with $F(v)$ as in Figure 3.4 and Equation (3.9) and with all of the masses m_n equal to 1. Notice that the RHS of (3.4) has a discontinuity at a point $\mathbf{u}$ if and only if $u_i = u_{i+1}$ for some i between 1 and $N-1$. It is when a trajectory reaches such a point $\mathbf{u}$ that we need the following definition. In terms of the spring-mass model of Figure 3.1, once the vertical positions u_i and u_{i+1} of two neighboring particles become equal, the spring connecting them is replaced by a rigid link. In other words,

---

[1] Having such a definition is crucial because, as we will show in Section 3.4, equation (3.4) will reach a discontinuity point of its RHS in finite time, starting with any initial condition.

the two particles are simply merged into a single particle which is twice as heavy (see Figure 3.5), yielding the following modification of (3.4) for $n = i$ and $n = i + 1$:

$$\dot{u}_i = \dot{u}_{i+1} = \frac{1}{2}(F(u_{i+2} - u_{i+1}) - F(u_i - u_{i-1})).$$

(The differential equations for $n \neq i, i+1$ do not change.) Similarly, if m consecutive particles reach equal vertical position, they are merged into one particle of mass m $(1 \leq m \leq N)$:

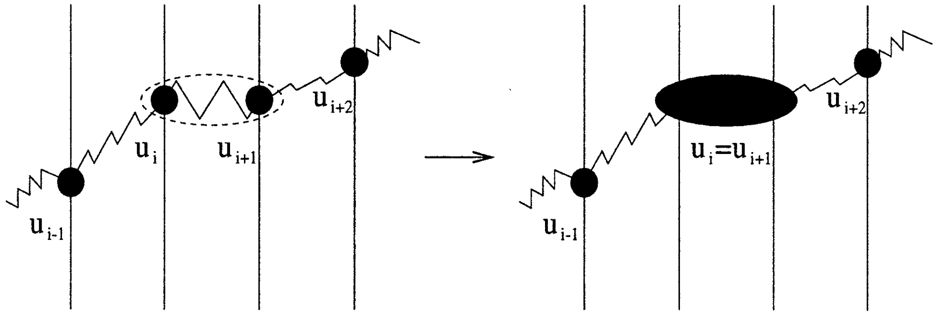


Figure 3.5. A horizontal spring is replaced by a rigid link.

$$\begin{aligned} \dot{u}_n &= \ldots = \dot{u}_{n+m-1} = \\ &= \frac{1}{m}(F(u_{n+m} - u_{n+m-1}) - F(u_n - u_{n-1})) \qquad (3.10) \\ \text{if} & \\ u_{n-1} &\neq u_n = u_{n+1} = \ldots = u_{n+m-2} = u_{n+m-1} \neq u_{n+m}. \end{aligned}$$

Notice that this system is the same as (3.4), but with possibly unequal masses. It is convenient to re-write this equation so as to explicitly indicate the reduction in the number of state variables:

$$\begin{aligned} &\dot{u}_{n_i} = \frac{1}{m_{n_i}}(F(u_{n_{i+1}} - u_{n_{i+1}-1}) - F(u_{n_i} - u_{n_{i-1}})), \qquad (3.11) \\ &u_{n_i} = u_{n_i+1} = \ldots = u_{n_i+m_{n_i}-1}, \\ &\text{where } i = 1, \ldots p, \\ &1 = n_1 < n_2 < \ldots < n_{p-1} < n_p \leq N, \\ &n_{i+1} = n_i + m_{n_i}. \end{aligned}$$

The compound particle described by the vertical position u_{n_i} and mass m_{n_i} consists of m_{n_i} unit-mass particles $u_{n_i}, u_{n_i+1}, \cdots, u_{n_i+m_{n_i}-1}$ that have been merged, as shown in Figure 3.5. The evolution can then naturally be thought of as a sequence of stages: during each stage, the right-hand side of (3.11) is continuous. Once the solution hits a discontinuity surface of the right-hand side, the state reduction and re-assignment of m_{n_i}'s, described above, takes place. The solution then proceeds according to the modified equation until it hits the next discontinuity surface, etc.

Notice that such an evolution automatically produces a multiscale segmentation of the original signal if one views each compound particle as a region of the signal. Viewed as a segmentation algorithm, this evolution can be summarized as follows:

1. Start with the trivial initial segmentation: each sample is a distinct region.
2. Evolve (3.11) until the values in two or more neighboring regions become equal.
3. Merge the neighboring regions whose values are equal.
4. Go to step 2.

The same algorithm can be used for 2-D images, which is immediate upon re-writing Equation (3.11):

$$\dot{u}_{n_i} = \frac{1}{m_{n_i}} \sum_{n_j \in A_{n_i}} F(u_{n_j} - u_{n_i}) p_{ij}, \tag{3.12}$$

where

m_{n_i} is again the mass of the compound particle n_i (= the number of pixels in the region n_i);

A_{n_i} is the set of the indices of all the neighbors of n_i, i.e., of all the compound particles that are connected to n_i by springs;

p_{ij} is the number of springs between regions n_i and n_j (always 1 in 1-D, but can be larger in 2-D).

Just as in 1-D, two neighboring regions n_1 and n_2 are merged by replacing them with one region n of mass $m_n = m_{n_1} + m_{n_2}$ and the set of neighbors $A_n = A_{n_1} \cup A_{n_2} \backslash \{n_1, n_2\}$.

We close this section by describing one of the basic and most important properties of these evolutions, namely that the evolution is stable but nevertheless behaves like an inverse diffusion. Notice that a force function $F(v)$ satisfying (3.9) can be represented as the sum of an inverse diffusion force $F_{id}(v)$ and a positive multiple of $\operatorname{sgn}(v)$: $F(v) = F_{id}(v) + C \operatorname{sgn}(v)$, where $C = F(0^+)$ and $-F_{id}(v)$ satisfies (3.3) and (3.6). Therefore, if $u_{n_{i+1}} - u_{n_i}$ and $u_{n_i} - u_{n_{i-1}}$ are of the same sign (which means that u_{n_i} is not a local extremum of the sequence $(u_{n_1}, \ldots, u_{n_p})$), then (3.11) can be written as

$$\dot{u}_{n_i} = \frac{1}{m_{n_i}} (F_{id}(u_{n_{i+1}} - u_{n_i}) - F_{id}(u_{n_i} - u_{n_{i-1}})). \tag{3.13}$$

If $u_{n_i} > u_{n_{i+1}}$ and $u_{n_i} > u_{n_{i-1}}$ (i.e., u_{n_i} is a local maximum), then (3.11) is

$$\dot{u}_{n_i} = \frac{1}{m_{n_i}} (F_{id}(u_{n_{i+1}} - u_{n_i}) - F_{id}(u_{n_i} - u_{n_{i-1}}) - 2C). \tag{3.14}$$

If $u_{n_i} < u_{n_{i+1}}$ and $u_{n_i} < u_{n_{i-1}}$ (i.e., u_{n_i} is a local minimum), then (3.11) is

$$\dot{u}_{n_i} = \frac{1}{m_{n_i}}(F_{id}(u_{n_{i+1}} - u_{n_i}) - F_{id}(u_{n_i} - u_{n_{i-1}}) + 2C). \tag{3.15}$$

Equation (3.13) says that the evolution is a pure inverse diffusion at the points which are not local extrema. It is not, however, a *global* inverse diffusion, since pure inverse diffusions drive local maxima to $+\infty$ and local minima to $-\infty$ and thus are unstable. In contrast, equations (3.14) and (3.15) show that at local extrema, the evolution introduced in this chapter is an inverse diffusion plus a stabilizing term which guarantees that the local maxima do not increase and the local minima do not decrease. Indeed, $|F_{id}(v)| \le F(0^+) = C$ for any v and for any SIDE force function F, and therefore the RHS of (3.14) is negative, and the RHS of (3.15) is positive. For this reason, we call the new evolution (3.11), (3.12) a "stabilized inverse diffusion equation" ("SIDE"), a force function satisfying (3.9) a "SIDE force", and the corresponding energy a "SIDE energy". In Chapter 4, we will analyze a simpler version of this equation, which results from dropping the inverse diffusion term. In this particular case, the local extrema move with constant speed and all the other samples are stationary, which makes the analysis of the equation more tractable.

3.4 Properties of SIDEs.

3.4.1 Basic Properties in 1-D.

The SIDEs described in the two preceding sections enjoy a number of interesting properties which validate and explain their adaptability to segmentation problems. We first examine the SIDEs in one spatial dimension for which we can make the strongest statements.

We define the n_i-th discontinuity hyperplane of a SIDE (3.11) by $S_{n_i} = \{\mathbf{u} \in \mathbb{R}^p : u_{n_i} = u_{n_{i+1}}\}$, $i = 1, \ldots, p-1$. Sometimes it is more convenient to work with the vector $\mathbf{v} = (v_{n_1}, \ldots, v_{n_{p-1}})^T \in \mathbb{R}^{p-1}$ of the first differences of $\mathbf{u}$: $v_{n_i} = u_{n_{i+1}} - u_{n_i}$, for $i = 1, \ldots, p-1$. We abuse notation by also denoting $S_{n_i} = \{\mathbf{v} \in \mathbb{R}^{p-1} : v_{n_i} = 0\}$.

On such hyperplanes, we defined the solution of a SIDE as the solution to a modified, lower-dimensional, equation whose RHS is continuous on S_{n_i}. In what follows, we will assume that the SIDE force function $F(v)$ is sufficiently regular away from zero, so that the ODE (3.11), restricted to the domain of continuity of its RHS, is well-posed. As a result, existence and uniqueness of solutions of SIDEs immediately follow from the existence and uniqueness of solutions of ODEs with continuous RHS. Continuous dependence on the initial data is also guaranteed for a trajectory segment lying inside a region of continuity of the RHS. In order to show, however, that the solutions that we have defined are continuous with respect to initial conditions over *arbitrary* time intervals, we must take into account the presence of discontinuities on the RHS. In particular, what must be shown is that trajectories that start very near a discontinuity surface remain close to one that starts on the surface. More precisely, we need to be able to show that a trajectory whose initial point is very close to S_{n_i} will, in fact, hit S_{n_i} (see Figure 3.6). In the literature on differential equations and control theory [17, 67], the behavior that SIDEs exhibit and which is illustrated in Figure 3.6 is referred to as "sliding modes". Specifically, as proven in Appendix A, the behavior of the evolution near discontinuity hyperplanes satisfies the following:

Lemma 3.1 (Lemma on Sliding). *Let σ be a permutation of $(n_1, \ldots, n_{p-1})$, and m an integer between 1 and $p-1$, and let S be the set of all points in the intersection of m hyperplanes which do not belong to the remaining $p-m-1$ hyperplanes:*

$$S = \bigcap_{q=1}^{m} S_{\sigma(q)} \setminus \left(\bigcup_{q=m+1}^{p-1} S_{\sigma(q)} \right).$$

Then, as $\mathbf{v}$ approaches S from any quadrant,[2] its velocity is directed towards S:

$$\lim(\dot{v}_{\sigma(q)}\ sign(v_{\sigma(q)})) \leq 0 \ for\ q = 1, \ldots, m,$$

and for at least one q this inequality is strict. ∎

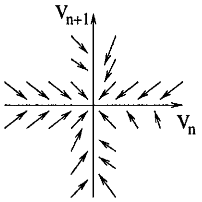


Figure 3.6. Solution field near discontinuity surfaces.

Intuitively, and as illustrated in Figure 3.6, this lemma states that the solution field of a SIDE near any discontinuity surface points toward that surface. As a consequence, a trajectory which hits such a surface may be continuously extended to "slide" along the surface, as shown in [17, 67]. For this reason the discontinuity surfaces are commonly referred to as "sliding surfaces". For SIDEs, a simple calculation verifies that the dynamics along such a surface, obtained through any of the three classical definitions in [17, 67], correspond exactly to the definition given in the preceding section.

The Lemma on Sliding, together with the well-posedness of SIDEs inside their continuity regions, directly implies the overall well-posedness of 1-D SIDEs: for finite T, the trajectory from $t = 0$ to $t = T$ depends continuously on its initial point. As shown in Property 3.2 to follow, a SIDE reaches a steady state in finite time, which establishes its well-posedness for infinite time intervals.

---

[2] In $\mathbb{R}^{p-1}$, a quadrant containing a vector $\mathbf{a} = (a_1, \ldots, a_{p-1})^T$ such that $a_i \neq 0$ for $i = 1, \ldots, p-1$ is the set $Q = \{\mathbf{b} \in \mathbb{R}^{p-1} : b_i a_i > 0 \text{ for } i = 1, \ldots, p-1\}$.

We call u_{n_i}, with $i \in \{2, \ldots, p-1\}$ a local maximum (minimum) of the sequence $(u_{n_1}, \ldots, u_{n_p})$ if $u_{n_i} > u_{n_{i\pm 1}}$ $(u_{n_i} < u_{n_{i\pm 1}})$. The point u_{n_1} is a local maximum (minimum) if $u_{n_1} > u_{n_2}$ $(u_{n_1} < u_{n_2})$; u_{n_p} is a local maximum (minimum) if $u_{n_p} > u_{n_{p-1}}$ $(u_{n_p} < u_{n_{p-1}})$. Therefore, we immediately have (as we saw in Equations (3.14), (3.15)) that the maxima (minima) are always pulled down (up):

Property 3.1 (maximum principle). *Every local maximum is decreased and every local minimum is increased by a SIDE. Therefore,*

$$|u_i(t)| < \max_n |u_n(0)| \text{ for } t > 0. \tag{3.16}$$

Using this result, we can prove the following:

Property 3.2 (finite evolution time). *A SIDE, started at* $\mathbf{u}^0 = (u_1^0, \ldots, u_N^0)^T$, *reaches its equilibrium (i.e., the point* $\mathbf{u} = (u_1, \ldots, u_N)^T$ *where* $u_1 = \ldots = u_N = \frac{1}{N}\sum_{i=1}^N u_i^0$*) in finite time.*

Proof. The sum of the vertical positions of all unit-mass particles is equal to the sum of the vertical positions of the compound particles, weighted by their masses: $\sum_{n=1}^N u_n = \sum_{i=1}^p u_{n_i} m_{n_i}$. The time derivative of this quantity is zero, as verified by summing up the right-hand sides of (3.11). Therefore, the mean vertical position $\frac{1}{N}\sum_{n=1}^N u_n$ is constant throughout the evolution. Writing (3.11) for $i = 1$, $\dot{u}_{n_1} = \frac{1}{m_{n_1}} F(u_{n_2} - u_{n_1})$, we see that the leftmost compound particle is stationary only if $p = 1$, i.e., if all unit-mass particles have the same vertical position: $u_{n_1} = u_1 = u_2 = \ldots = u_N$. Since the mean is conserved, the unique steady state is $u_1 = \ldots = u_N = \frac{1}{N}\sum_{i=1}^N u_i^0$. To prove that it is reached in finite time, we again refer to the spring-mass model of Figure 3.1 and use the fact that a SIDE force function assigns larger force to shorter springs. If we put $L = 2 \max_n |u_n(0)|$, then the maximum principle implies that in the system there cannot exist a spring with vertical extent larger than L at any time during the evolution. Therefore, the rate of decrease of the absolute maximum, according to Equation (3.11), is at least $F(L)/N$ (because $F(L)$ is the smallest force possible in the system, and N is the largest mass). Similarly, the absolute minimum always increases at least as quickly. They will meet no later than at $t = \frac{LN}{2F(L)}$, at which point the sequence $\mathbf{u}(t)$ must be a constant sequence. ■

The above property allows us immediately to state the well-posedness results as follows:

Property 3.3 (well-posedness). *For any initial condition* $\mathbf{u}^{0*}$, *a SIDE has a unique solution* $\mathbf{u}^*(t)$ *satisfying* $\mathbf{u}^*(0) = \mathbf{u}^{0*}$. *Moreover, for any such* $\mathbf{u}^{0*}$ *and any* $\varepsilon > 0$, *there exists a* $\delta > 0$ *such that* $|\mathbf{u}^0 - \mathbf{u}^{0*}| \le \delta$ *implies* $|\mathbf{u}(t) - \mathbf{u}^*(t)| \le \varepsilon$ *for* $t \ge 0$, *where* $\mathbf{u}(t)$ *is the solution of the SIDE with the initial condition* $\mathbf{u}^0$. ■

As we pointed out in the introductory section of this chapter, a SIDE evolution defines a natural set of hitting times which intuitively should be of use in characterizing features in an image. For this to be true, however, we would need some type of continuity of this hitting time sequence. Specifically, let $t_n(\mathbf{u}^0)$ denote the "n-th hit time", i.e., the time when the solution starting at $\mathbf{u}^0$ reaches the sliding hyperplane S_n. By Property 3.2, this is a finite number. Let $\mathbf{u}(t)$ be "a typical solution" if it never reaches two different sliding hyperplanes at the same time: $t_i(\mathbf{u}(0)) \neq t_j(\mathbf{u}(0))$ if $i \neq j$. One of the consequences of the Lemma on Sliding is that a trajectory that hits a single hyperplane S_n does so transversally (that is, cannot be tangent to it). Since trajectories vary continuously, this means that nearby solutions also hit S_n. Therefore, for typical solutions the following holds:

Property 3.4 (stability of hit times). *If $\mathbf{u}(t)$ is a typical solution, all solutions with initial data sufficiently close to $\mathbf{u}(0)$ get onto surfaces S_n in the same order as $\mathbf{u}(t)$.* ■

The sequence in which a trajectory hits surfaces S_n is an important characteristic of the solution. Property 3.4 says that, for a typical solution $\mathbf{u}(t)$, the (strict) ordering of hit times $t_n(\mathbf{u}(0))$ is stable with respect to small disturbances in $\mathbf{u}(0)$:

$$t_{n_1}(\mathbf{u}(0)) < t_{n_2}(\mathbf{u}(0)) < \ldots < t_{n_{N-1}}(\mathbf{u}(0)), \tag{3.17}$$

where $(n_1, \ldots, n_{N-1})$ is a permutation of $(1, \ldots, N-1)$. For the purposes of segmentation and edge detection, the only interesting output occurs at these $N-1$ time points, since they are the only instants when the segmentation of the initial signal changes (i.e., when regions are merged and edges are erased). While a thorough investigation of how to use these hitting times and in particular how to stop a SIDE so as to obtain the best segmentation is an open one for the general form of the SIDE force function F, we will obtain a partial answer for a certain choice of F in the next chapter. Specifically, when the number of regions is an exponential random variable, and $F(v) = \text{sgn}(v)$, then the stopping rule in 1-D is given by Proposition 4.6 of the next chapter. For other SIDE force functions, the fact that the choice of the output time points is limited to a finite set provides us with both a natural sequence of segmentations of increasing granularity and with, at the very least, some simple stopping rules. For example, if the number of "useful" regions, r, is known or bounded a priori, a natural candidate for a stopping time would be $t_{n_{N-r}}$, i.e., the time when exactly r regions remain. In the next section we illustrate the effectiveness of such a rule in the simplest case, namely when $r = 2$ so that we are seeking a partition of the field of interest into two regions.

We already mentioned that our definition of solutions on sliding surfaces for SIDEs in one spatial dimension coincides with all three classical definitions of solutions for a general equation with discontinuous right-hand side, which are presented on pages 50-56 of Filippov's book [17]. We use a result on page 95 of [17] to infer the following:

Property 3.5 (continuous dependence on the RHS). *Let us consider a SIDE force function $F_S(v)$, and let $p_K(v)$ be a smoothing kernel of width K:*

$$p_K(v) \geq 0, \quad supp(p_K) = [-K; K], \quad \int p_K(v)\, dv = 1.$$

Let $F_K(v) = \int F_S(w)p_K(v-w)\,dw$ *be a regularized version of* $F_S(v)$. *Consider system (3.4) with* $m_n = 1$ *and* $F(v) = F_K(v)$. *Then for any* ε, *there is a* K *such that the solution of this system stays closer than* ε *to the solution of the SIDE with the same initial condition and force* $F_S(v)$. ■

We note that if the smoothing kernel $p_K(v)$ is appropriately chosen, then the resulting $F_K(v)$ will be a Perona-Malik force function of thickness K. (For example, one easy choice for $p_K(v)$ is a multiple of the indicator function of the interval [-K;K].) Thus, semi-discrete Perona-Malik evolutions with small K are regularizations of SIDEs, and consequently a SIDE in 1-D can be viewed as a limiting case of a Perona-Malik-type evolution. However, as we will see in the experimental section, the SIDE evolutions appear to have some advantages over such regularized evolutions even in 1-D.

■ 3.4.2 Energy Dissipation in 1-D.

It was mentioned in the previous chapter that the SIDE (3.11) is the gradient descent equation for the global energy

$$\mathcal{E}(\mathbf{u}) = \sum_{n=1}^{N-1} E(u_{n+1} - u_n), \tag{3.18}$$

where E is the SIDE energy function (Figure 2.6), i.e., an antiderivative of the SIDE force function. Note that the standard definition of the gradient cannot be used here. Indeed, non-differentiability of E at the origin makes the directional derivatives of $\mathcal{E}(\mathbf{u})$ in the directions orthogonal to a sliding surface S undefined for $\mathbf{u} \in S$. But once $\mathbf{u}(t)$ hits a sliding surface, it stays there for all future times, and therefore we do not have to be concerned with the partial derivatives of $\mathcal{E}(\mathbf{u})$ in the directions which do not lie in the sliding surface. This leads to the definition of the gradient as the vector of partial derivatives taken with respect to the directions which belong to the sliding surface.

Definition 3.1. *Suppose that* S *is the intersection of all the sliding hyperplanes of the SIDE (3.11) to which the vector* $\mathbf{u}$ *belongs. Suppose further that* $\{\mathbf{f}_i\}_{i=1}^p$ *is an orthonormal basis for* S. *Then the gradient of* $\mathcal{E}$ *with respect to* S, $\nabla_S\mathcal{E}$, *is defined as the weighted sum of the basis vectors, with the weights equal to the corresponding directional derivatives:*

$$\nabla_S\mathcal{E}(\mathbf{u}) \overset{\text{def}}{=} \sum_{i=1}^{p} \frac{\partial\mathcal{E}(\mathbf{u})}{\partial\mathbf{f}_i}\mathbf{f}_i. \tag{3.19}$$

We will show in this section that at any moment t, the RHS of the SIDE (3.11) is the negative gradient of $\mathcal{E}(\mathbf{u}(t))$, taken with respect to the intersection S of all the sliding surfaces to which $\mathbf{u}(t)$ belongs. An auxiliary result is needed in order to show this.

Lemma 3.2. *Suppose that, as in Equation (3.11),* $\mathbf{u}$ *is a signal with* p *distinct regions of masses* $m_1, \ldots, m_p$*:*

$$\begin{array}{rll} 1 = n_1 < n_2, \ldots < n_{p-1} < n_p \leq N & \text{are such that} & \\ n_{i+1} = n_i + m_{n_i}, & \text{and} & (3.20) \\ u_{n_i - 1} \neq u_{n_i} = u_{n_i+1} = \ldots = u_{n_{i+1}-1} \neq u_{n_{i+1}}, & \text{for } i = 1, \ldots, p. & \end{array}$$

Let $\{\mathbf{e}_j\}_{j=1}^N$ *be the standard basis of* $I\!\!R^N$ *(i.e., the* j*-th entry of* $\mathbf{e}_j$ *is* 1 *and all other entries are zeros), and define*

$$\mathbf{f}_i = \frac{1}{\sqrt{m_{n_i}}} \sum_{j=n_i}^{n_{i+1}-1} \mathbf{e}_j, \text{ for } i = 1, \ldots, p. \tag{3.21}$$

Then $\{\mathbf{f}_i\}_{i=1}^p$ *is an orthonormal basis for the sliding surface* S *defined by (3.20).*

Proof. The vector $\mathbf{f}_i$ satisfies Equation (3.20), and therefore it belongs to the sliding surface S. Since $\mathbf{e}_j$'s are mutually orthogonal, so are $\mathbf{f}_i$'s. Since there are p distinct $\mathbf{f}_i$'s, they form a basis for the p-dimensional surface S. The norm of $\mathbf{f}_i$ is

$$\sum_{j=n_i}^{n_{i+1}-1} \left(\frac{1}{\sqrt{m_{n_i}}} \right)^2 = m_{n_i} \frac{1}{m_{n_i}} = 1. \qquad \blacksquare$$

Property 3.6 (gradient descent). *The SIDE (3.11) is the gradient descent equation for the global energy (3.18), i.e.,*

$$\dot{\mathbf{u}}(t) = -\nabla_{S(t)} \mathcal{E}(\mathbf{u}(t)), \tag{3.22}$$

where $S(t)$ *is the intersection of all sliding hyperplanes to which* $\mathbf{u}(t)$ *belongs, and* $\nabla_{S(t)}$ *is the gradient with respect to* $S(t)$*.*

Proof. In order to prove this property, we write out Equation (3.22) in terms of the coefficients of $\dot{\mathbf{u}}$ and $-\nabla_S \mathcal{E}(\mathbf{u})$ with respect to the basis $\{\mathbf{f}_i\}_{i=1}^p$ (3.21). It is immediate from the definition (3.21) of $\mathbf{f}_i$'s that

$$\mathbf{u} = \sum_{i=1}^p u_{n_i} \sqrt{m_{n_i}} \mathbf{f}_i,$$

and so the i-th coefficient of $\dot{\mathbf{u}}$ in the basis $\{\mathbf{f}_i\}_{i=1}^p$ is

$$\sqrt{m_{n_i}} \dot{u}_{n_i} \tag{3.23}$$

Since the basis $\{\mathbf{f}_i\}_{i=1}^p$ is orthonormal, the i-th coefficient of $-\nabla_S \mathcal{E}(\mathbf{u})$ in this basis is the directional derivative of $-\mathcal{E}$ in the direction $\mathbf{f}_i$:

$$-\frac{\partial \mathcal{E}}{\partial \mathbf{f}_i} = -\lim_{\Delta \to 0} \frac{1}{\Delta} \{\mathcal{E}(\mathbf{u} + \mathbf{f}_i \Delta) - \mathcal{E}(\mathbf{u})\}$$

$$\begin{aligned}
&= -\lim_{\Delta\to 0}\frac{1}{\Delta}\left\{\left[\sum_{n=1}^{n_i-2}E(u_{n+1}-u_n)+E(u_{n_i}+\frac{\Delta}{\sqrt{m_{n_i}}}-u_{n_i-1})\right.\right.\\
&\quad+\sum_{n=n_i}^{n_{i+1}-2}E(u_{n+1}+\frac{\Delta}{\sqrt{m_{n_i}}}-u_n-\frac{\Delta}{\sqrt{m_{n_i}}})\\
&\quad\left.\left.+E(u_{n_{i+1}}-u_{n_{i+1}-1}-\frac{\Delta}{\sqrt{m_{n_i}}})+\sum_{n=n_{i+1}}^{N-1}E(u_{n+1}-u_n)\right]-\sum_{n=1}^{N-1}E(u_{n+1}-u_n)\right\}\\
&= -\lim_{\Delta\to 0}\left\{\frac{1}{\Delta}\left[E(u_{n_i}-u_{n_i-1}+\frac{\Delta}{\sqrt{m_{n_i}}})-E(u_{n_i}-u_{n_i-1})\right]\right.\\
&\quad\left.+\frac{1}{\Delta}\left[E(u_{n_{i+1}}-u_{n_{i+1}-1}-\frac{\Delta}{\sqrt{m_{n_i}}})-E(u_{n_{i+1}}-u_{n_{i+1}-1})\right]\right\}\\
&= -\left\{\frac{1}{\sqrt{m_{n_i}}}E'(u_{n_i}-u_{n_i-1})-\frac{1}{\sqrt{m_{n_i}}}E'(u_{n_{i+1}}-u_{n_{i+1}-1})\right\}\\
&= \frac{1}{\sqrt{m_{n_i}}}(F(u_{n_{i+1}}-u_{n_{i+1}-1})-F(u_{n_i}-u_{n_i-1})). \qquad (3.24)
\end{aligned}$$

Equating the coefficients (3.23) and (3.24), we get that the gradient descent equation (3.22), written in the basis $\{\mathbf{f}_i\}_{i=1}^{p}$, is:

$$\dot{u}_{n_i}=\frac{1}{m_{n_i}}(F(u_{n_{i+1}}-u_{n_{i+1}-1})-F(u_{n_i}-u_{n_i-1}),$$

which is the SIDE (3.11). ■

It is possible to characterize further the process of energy dissipation during the evolution of a SIDE. Namely, between any two consecutive mergings (i.e., hits of a sliding surface), the energy is a concave function of time.

Property 3.7 (energy dissipation). *Consider the SIDE (3.11) and let E be the corresponding SIDE energy function: $E'=F$. Then between any two consecutive mergings during the evolution of the SIDE, the global energy (3.18) is decreasing and concave as a function of time:*

$$\begin{aligned}\dot{\mathcal{E}} &< 0\\ \ddot{\mathcal{E}} &\le 0.\end{aligned}$$

Proof. To simplify notation, we will denote

$$y_i=u_{n_i}\text{ for }i=1,\dots,p,$$

and will simply write m_i instead of m_{n_i}. Then the global energy (3.18) is

$$\mathcal{E}=\sum_{i=1}^{p}E(y_{i+1}-y_i),$$

and therefore the SIDE (3.11) can be re-written as follows:

$$\dot{y}_i = -\frac{1}{m_i}\frac{\partial \mathcal{E}}{\partial y_i}, \quad i = 1, \ldots, p.$$

By the chain rule of differentiation, we have:

$$\dot{\mathcal{E}} = \sum_{i=1}^{p} \frac{\partial \mathcal{E}}{\partial y_i}\dot{y}_i = -\sum_{i=1}^{p}\left(\frac{\partial \mathcal{E}}{\partial y_i}\right)^2 \frac{1}{m_i} < 0.$$

Differentiating with respect to t one more time and applying the chain rule again yields:

$$\begin{aligned}
\ddot{\mathcal{E}} &= -\sum_{i=1}^{p} 2\frac{\partial \mathcal{E}}{\partial y_i}\frac{d}{dt}\left(\frac{\partial \mathcal{E}}{\partial y_i}\right)\frac{1}{m_i} \\
&= -\sum_{i=1}^{p} 2\frac{\partial \mathcal{E}}{\partial y_i}\left(\sum_{k=1}^{p}\frac{\partial^2 \mathcal{E}}{\partial y_i \partial y_k}\dot{y}_k\right)\frac{1}{m_i} \\
&= \sum_{i=1}^{p} 2\frac{\partial \mathcal{E}}{\partial y_i}\left(\sum_{k=1}^{p}\frac{\partial^2 \mathcal{E}}{\partial y_i \partial y_k}\frac{1}{m_k}\frac{\partial \mathcal{E}}{\partial y_k}\right)\frac{1}{m_i} \\
&= 2D^T H D, \qquad (3.25)
\end{aligned}$$

where

$$D = \left(\frac{1}{m_1}\frac{\partial \mathcal{E}}{\partial y_1}, \ldots, \frac{1}{m_p}\frac{\partial \mathcal{E}}{\partial y_p}\right)^T,$$

and H is the Hessian matrix of $\mathcal{E}$, i.e., the matrix of all the mixed second derivatives of $\mathcal{E}$. The entry in the i-th row and k-th column of H is $\frac{\partial^2 \mathcal{E}}{\partial y_i \partial y_k}$. In other words,

$$H = -\begin{pmatrix}
x_1 & -x_1 & 0 & 0 & 0 & \ldots & & 0 \\
-x_1 & x_1 + x_2 & -x_2 & 0 & 0 & \ldots & & 0 \\
0 & -x_2 & x_2 + x_3 & -x_3 & 0 & \ldots & & 0 \\
\vdots & & & & & & & \vdots \\
 & & & & \ddots & \ddots & \ddots & 0 \\
0 & & \ldots & & 0 & -x_{p-2} & x_{p-2} + x_{p-1} & -x_{p-1} \\
0 & & \ldots & & & 0 & -x_{p-1} & x_{p-1}
\end{pmatrix},$$

where $x_i = -F'(y_{i+1} - y_i)$. Note that, by our definition of y_i's, $y_{i+1} - y_i \neq 0$, and that $F(y_{i+1} - y_i)$ is monotonically decreasing for $y_{i+1} - y_i \neq 0$. Therefore, $x_i > 0$.

All that remains to show is that H is negative semidefinite, which, combined with (3.25), means that $\ddot{\mathcal{E}} \leq 0$. It is easily verified that $-H$ can be factorized into a lowertriangular and an upper-triangular matrix as follows:

$$-H = \begin{pmatrix} 1 & 0 & & \dots & & 0 \\ -1 & 1 & 0 & \dots & & 0 \\ 0 & -1 & 1 & \dots & & 0 \\ \vdots & & & & & \vdots \\ & & \ddots & \ddots & \ddots & 0 \\ 0 & & \dots & -1 & 1 & 0 \\ 0 & & \dots & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 & -x_1 & 0 & 0 & \dots & & 0 \\ 0 & x_2 & -x_2 & 0 & \dots & & 0 \\ 0 & 0 & x_3 & -x_3 & \dots & & 0 \\ \vdots & & & & & & \vdots \\ & & & & \ddots & \ddots & 0 \\ 0 & & \dots & & & x_{p-1} & -x_{p-1} \\ 0 & & \dots & & & & 0 \end{pmatrix}.$$

The diagonal entries $x_1, \dots, x_{p-1}, 0$ of the upper-triangular matrix are the pivots ([62], page 32) of $-H$. Since all the pivots are nonnegative, it follows ([62], page 339) that $-H \geq 0 \Rightarrow H \leq 0$, which implies $\ddot{\mathcal{E}} \leq 0$. ■

A typical picture of the energy dissipation is shown in Figure 3.7; the only points where $\mathcal{E}$ might not be concave as a function of time are the merge points.

Figure 3.7. Typical picture of the energy dissipation during the evolution of a SIDE.

In addition to being the gradient descent equation for the global energy (3.18), with $E' = F$, we now show that the SIDE (3.11) also reduces $\sum_{n=1}^{N} E_1(u_{n+1} - u_n)$ if $E_1 \neq E$ is *any* SIDE energy function.

Property 3.8 (Lyapunov functionals). *Consider the SIDE (3.11), and let E be the corresponding SIDE energy function:* $E' = F$. *Let* E_1 *be an arbitrary SIDE energy function (i.e., such function that* E_1' *satisfies (3.3),(3.9)), and define*

$$\mathcal{E}_1(\mathbf{u}) = \sum_{n=1}^{N} E_1(u_{n+1} - u_n).$$

Then $\mathcal{E}_1$ *is a Lyapunov functional of the SIDE. In other words,* $\mathcal{E}_1(\mathbf{u}(t))$ *is a decreasing function of time, until the steady state is reached.*

Proof. We again use the notation from the proof of the previous property:

$$\begin{aligned} y_i &= u_{n_i} \text{ for } i = 1, \dots, p; \\ \mathcal{E}_1 &= \sum_{i=1}^{p} E_1(y_{i+1} - y_i). \end{aligned}$$

Then, by the chain rule,

$$\begin{aligned} \dot{\mathcal{E}} &= \sum_{i=1}^{p} \frac{\partial \mathcal{E}}{\partial y_i} \dot{y}_i \\ &= -E_1'(y_2 - y_1)\dot{y}_1 + \sum_{i=2}^{p-1} (E_1'(y_i - y_{i-1}) - E_1'(y_{i+1} - y_i))\dot{y}_i + E_1'(y_p - y_{p-1})\dot{y}_p \\ &= -\left[\frac{1}{m_1} E_1'(y_2 - y_1) F(y_2 - y_1) \right. \\ &\quad + \sum_{i=2}^{p-1} \frac{1}{m_i} (E_1'(y_i - y_{i-1}) - E_1'(y_{i+1} - y_i))(F(y_i - y_{i-1}) - F(y_{i+1} - y_i)) \\ &\quad \left. + \frac{1}{m_p} E_1'(y_p - y_{p-1}) F(y_p - y_{p-1}) \right]. \end{aligned}$$

The first term inside the brackets is positive, since $E_1'(y_2 - y_1)$, $F(y_2 - y_1)$, and $y_2 - y_1$ all have the same sign. Similarly, the last term is positive. Each term in the summation is also positive, because of monotonicity of E_1' and F. Therefore, $\dot{\mathcal{E}} < 0$. ■

We now analyze another class of Lyapunov functionals, which includes the ℓ^2 norm and negative entropy.

Property 3.9 (Lyapunov functionals, continued). *Suppose that $R : I\!R \to I\!R$ is a function such that its derivative R' is monotonically increasing. Define*

$$\mathcal{R}(\mathbf{u}) = \sum_{n=1}^{N} R(u_n).$$

Then $\mathcal{R}$ is a Lyapunov functional of Equation (3.4), i.e.,

$$\dot{\mathcal{R}} < 0,$$

until the steady state is reached. In particular, $\mathcal{R}$ is a Lyapunov functional of SIDEs.

Proof. Using the notation of the previous proof,

$$\dot{\mathcal{R}} = \frac{d}{dt} \left(\sum_{i=1}^{p} m_i R(y_i) \right)$$

$$\begin{aligned} &= \sum_{i=1}^{p} m_i R'(y_i)\dot{y}_i \\ &= R'(y_1)F(y_2-y_1) + \sum_{i=2}^{p-1} R'(y_i)(F(y_{i+1}-y_i) - F(y_i - y_{i-1})) - R'(y_p)F(y_p - y_{p-1}) \\ &= \sum_{i=1}^{p-1}(R'(y_i) - R'(y_{i+1}))F(y_{i+1}-y_i). \qquad (3.26) \end{aligned}$$

Since R' is monotonically increasing,

$$\operatorname{sgn}(R'(y_i) - R'(y_{i+1})) = \operatorname{sgn}(y_i - y_{i+1}) = -\operatorname{sgn}(y_{i+1} - y_i);$$

since F is a force function,

$$\operatorname{sgn}(F(y_{i+1} - y_i)) = sgn(y_{i+1} - y_i).$$

Therefore, the product $(R'(y_i) - R'(y_{i+1}))F(y_{i+1} - y_i)$ is negative, the sum (3.26) is negative, and so $\dot{\mathcal{R}} < 0$. ■

Example 3.3. *ℓ^q norms.*

Suppose $R(u_n) = |u_n|^q$ with $q > 1$. Then $R'(u_n) = \operatorname{sgn}(u_n)q|u_n|^{q-1}$ is monotonically increasing, which means that the ℓ^q norm

$$(\mathcal{R}(\mathbf{u}))^{\frac{1}{q}} = \left(\sum_{i=1}^{q} |u_n|^q\right)^{\frac{1}{q}}$$

is a Lyapunov functional of Equation (3.4). This is yet another characterization of the stability of the equations described by the mechanical model of Section 3.2. ■

Example 3.4. *Moments.*

It is similarly shown that the even central moments of $\mathbf{u}$ are also Lyapunov functionals:

$$\frac{1}{N}\sum_{n=1}^{N}\left(u_n - \frac{1}{N}\sum_{i=1}^{N} u_i\right)^{2k}, \ k = 1, 2, \ldots \quad ■$$

Example 3.5. *Entropy.*

Suppose the initial condition $\mathbf{u}^0$ is positive:

$$\min_{1 \le n \le N} u_n^0 > 0,$$

and define $R(u_n) = u_n \ln u_n$. Then $R'(u_n) = \ln u_n + 1$ is monotonically increasing, and so the negative entropy $\mathcal{R}(\mathbf{u}) = \sum_{n=1}^{N} u_n \ln u_n$ is a Lyapunov functional. The fact that the entropy is increased by SIDEs and other evolution equations of the form (3.4) is in agreement with the intuitive notion that, as scale increases, the signal is simplified: at coarser scales, the information content is reduced. ■

3.4.3 Properties in 2-D.

The existence and uniqueness of solutions in 2-D again follow easily from our construction of solutions. Property 3.1 (the maximum principle) is easily inferred from the 2-D spring-mass model. (A local maximum (minimum) is a region of a 2-D image whose value is larger (smaller) than the values of its neighbors. Re-phrasing this definition in terms of our spring-mass model, a maximum (minimum) is a particle with all its attached springs directed downward (upward).) Property 3.2 (finite evolution time) also carries over, with the same proof.

There is, however, no analog of the Lemma on Sliding in 2-D for SIDE force functions such as that depicted in Figure 3.4: it is easy to show that the solutions in the vicinity of a discontinuity hyperplane of (3.12) do not necessarily slide onto that hyperplane. Notice, however, that forcing two neighboring regions with equal intensities to merge is conceptually very similar to using a modified force function which is infinite at zero, as depicted in Figure 3.8. Indeed, the fact that $F(0^{\pm}) = \pm\infty$ means that if the vertical distance between two particles is very small compared to the distances to their other neighbors, they will be driven towards each other and will be merged. Thus, the Lemma on Sliding holds for the force function of Figure 3.8, from which the global continuous dependence on the initial data is again inferred. We do not use this force function in simulations, since its large values near zero present problems with numerical integration of the corresponding equation. What we do use is a SIDE force function, in conjunction with Equation (3.12). Since sliding modes do not necessarily occur on the discontinuity hyperplanes, there is no global continuous dependence on the initial data. In particular, the sequence of hitting times and associated discontinuity planes does not depend continuously on initial conditions, and our SIDE evolution does not correspond to a limiting form of a Perona-Malik evolution in 2-D but in fact represents a decidedly different type of evolutionary behavior. Several factors, however, indicate the value of this new evolution and also suggest that a weaker stability result can be proven. First of all, as shown in the experimental results in the next section, SIDEs can produce excellent segmentations in 2-D images even in the presence of considerable noise. Moreover, thanks to the maximum principle, excessively wild behavior of solutions is impossible, something that is again confirmed by the experiments of the next section. Consequently, the sequence of hit times (3.17) does not seem to be very sensitive to the initial condition in that the presence of noise, while perhaps perturbing the ordering of hitting times and the sliding planes that are hit, seems to introduce perturbations that are, in some sense, "small".

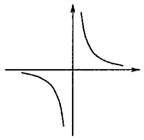


Figure 3.8. A modified force function, for which sliding happens in 2-D, as well as in 1-D.

Finally, we note without giving details that the properties on energy dissipation (3.6 and 3.7) and Property 3.9 on Lyapunov functionals, carry over to 2-D, as well as their proofs—with slight changes to accommodate the fact that a region may have more than two neighbors in 2-D.

3.5 Experiments.

In this section we present examples in both 1-D and 2-D. The purpose of 1-D experiments is to provide the basic intuition for how SIDEs work, as well as to contrast SIDEs with the methods reviewed in the previous chapter. We do not claim that SIDEs are the best for any of these 1-D examples, for which good results can be efficiently obtained using simple algorithms. In 2-D, however, this is no longer true, and SIDEs have considerable advantages over the existing methods.

Choosing a SIDE force function best suited for a particular application is an open research question. (It is partly addressed in Chapter 4, by describing the problems for which $F(v) = \mathrm{sgn}(v)$ is the best choice.) For the examples below, we use a very simple, piecewise-linear force function $F(v) = \mathrm{sgn}(v) - \frac{v}{L}$, depicted in Figure 3.9. Note that, formally, this function does not satisfy our definition (3.3) of a force function, since it is negative for $v > L$. Therefore, in our experiments we always make sure that L is larger than the dynamic range of the signal or image to be processed. In that case, thanks to the maximum principle, we will have $|u_i(t) - u_j(t)| < L$ for any pair of pixels at any time t during evolution, and therefore $F(|u_i(t) - u_j(t)|) > 0$.

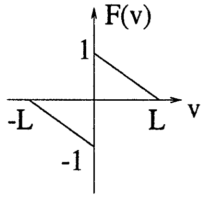
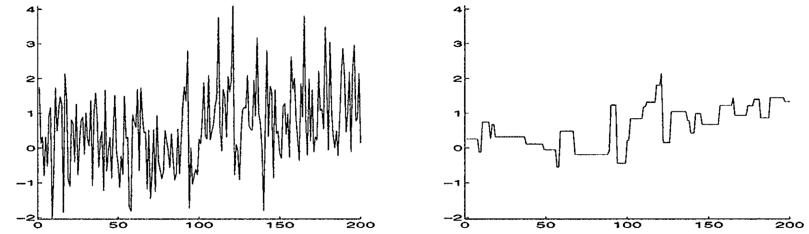


Figure 3.9. The SIDE force function used in the experimental section.

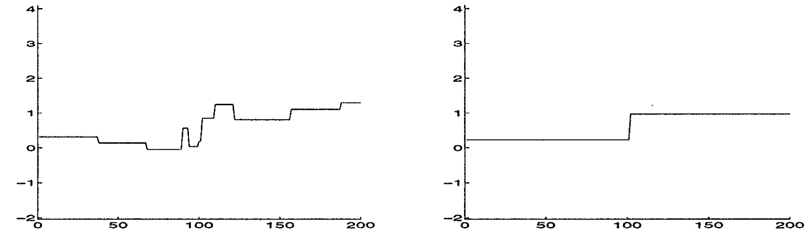
As we mentioned before, choosing the appropriate stopping rule is also an open problem. In the examples to follow, we assume that we know the number of regions we are looking for, and stop the evolution when that number of regions is achieved.

3.5.1 Experiment 1: 1-D Unit Step in High Noise Environment.

We first test this SIDE on a unit step function corrupted by additive white Gaussian noise whose standard deviation is equal to the amplitude of the step, and which is depicted in Figure 3.10(a). The remaining parts of this figure display snapshots of the SIDE evolution starting with the noisy data in Figure 3.10(a), i.e., they correspond to the evolution at a selected set of hitting times. The particular members of the scale space which are illustrated are labeled according to the number of remaining regions.



(a) Initial signal: noisy unit step. (b) 33 regions left.



(c) 11 regions left. (d) 2 regions left.

Figure 3.10. Scale space of a SIDE for a noisy unit step at location 100: (a) the original signal; (b)–(d) representatives of the resulting SIDE scale space.

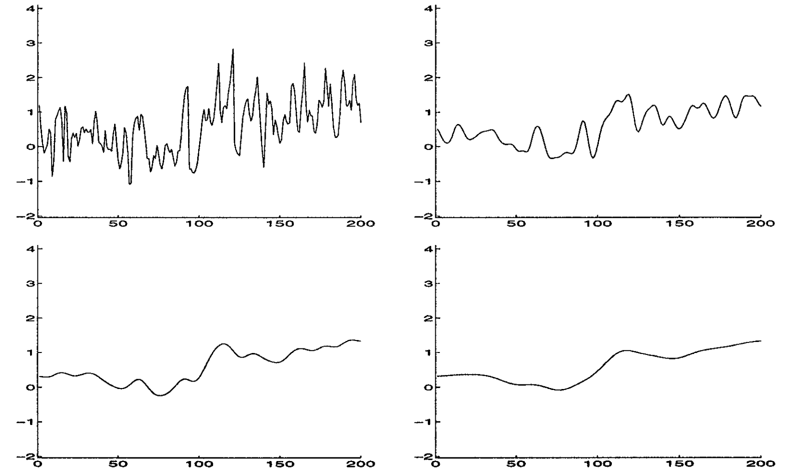
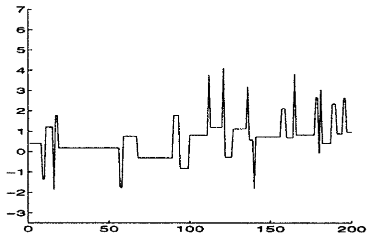
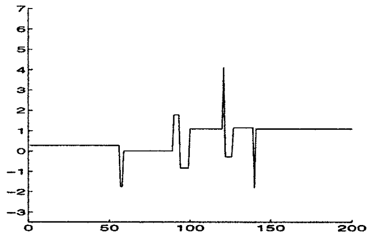


Figure 3.11. Scale space of a Perona-Malik equation with a large K for the noisy step of Figure 3.10.

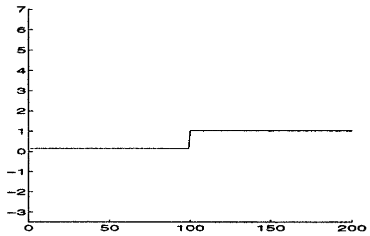
Note that the last remaining edge, i.e., the edge in Figure 3.10(d) for the hitting time at which there are only two regions left, is located between samples 101 and 102, which is quite close to the position of the original edge (between the 100-th and 101-st samples). In this example, the step in Figure 3.10(d) also has amplitude that is close to that of the original unit step. In general, thanks to the stability of SIDEs, the sizes of discontinuities will be diminished through such an evolution, much as they are in other evolution equations. However, from the perspective of segmentation this is irrelevant–



(a) 33 regions left.



(b) 11 regions left.



(c) 2 regions left.

Figure 3.12. Scale space of the region merging algorithm of Koepfler, Lopez, and Morel for the noisy unit step signal of Figure 3.10(a).

i.e., the focus of attention is on detecting and locating the edge, not on estimating its amplitude.

This example also provides us with the opportunity to contrast the behavior of a SIDE evolution with a Perona-Malik evolution and in fact to describe the behavior that originally motivated our work. Specifically, as we noted in the discussion of Property 3.5 of the previous section, a SIDE in 1-D can be approximated with a Perona-Malik equation of a small thickness K. Observe that a Perona-Malik equation of a large thickness K will diffuse the edge before removing all the noise. Consequently, if the objective is segmentation, the desire is to use as small a value of K as possible. Following the procedure prescribed by Perona, Shiota, and Malik in [50], we computed the histogram of the absolute values of the gradient throughout the initial signal, and fixed K at 90% of its integral. The resulting evolution is shown in Figure 3.11. In addition to its good denoising performance, it also blurs the edge, which is clearly undesirable if the objective is a sharp segmentation. The comparison of Figures 3.10 and 3.11 strongly suggests that the smaller K the better. It was precisely this observation that originally

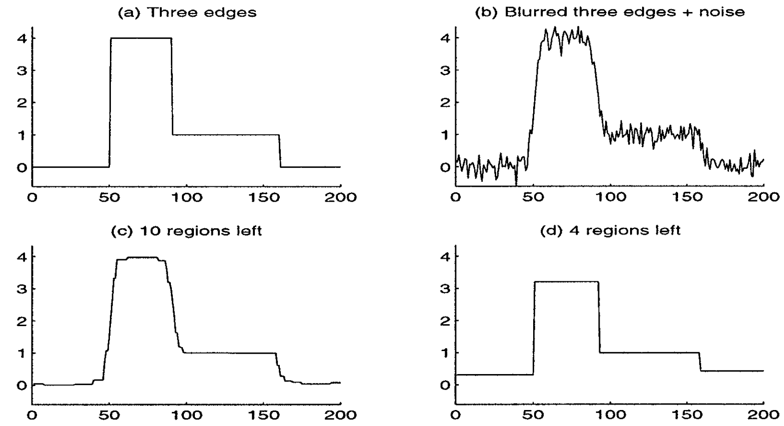


Figure 3.13. Scale space of a SIDE for a noisy blurred 3-edge staircase: (a) noise-free original signal; (b) its blurred version with additive noise; (c),(d) representatives of the resulting SIDE scale space.

motivated the development of SIDEs. However, while in 1-D a SIDE evolution can be viewed precisely as a limit of a Perona-Malik evolution as K goes to 0, there is still an advantage to using the form of the evolution that we have described rather than a Perona-Malik evolution with a very small value of K. Specifically, the presence of explicit reductions in dimensionality during the evolution makes a SIDE implementation more efficient than that described in [50]. Even for this simple example the Perona-Malik evolution that produced the result comparable to that in Figure 3.10 evolved approximately 5 times more slowly than our SIDE evolution. (Both were implemented via forward Euler discretization schemes [14] in MATLAB.) Although a SIDE in 2-D cannot be viewed as a limit of Perona-Malik evolutions, the same comparison in speed of evolution is still true, although in this case the difference in computation time can be orders of magnitude.

In this example, the region merging method of Koepfler, Lopez and Morel [36] works quite well (see Figure 3.12). We will soon see, however, that it is not as robust as SIDEs: its performance worsens dramatically when signals are corrupted with a heavy-tailed noise.

3.5.2 Experiment 2: Edge Enhancement in 1-D.

Our second one-dimensional example shows that SIDEs can stably enhance edges. The staircase signal in the upper left-hand corner of Figure 3.13 was convolved with a Gaussian and corrupted by additive noise. The evolution was stopped when there were only four regions (three edges) left. The locations of the edges are very close to those in the original signal. (Note that the amplitudes of the final signal are quite different from those of the initial condition. This is immaterial, since we are interested in segmentation, not in restoration.)

■ 3.5.3 Experiment 3: Robustness in 1-D.

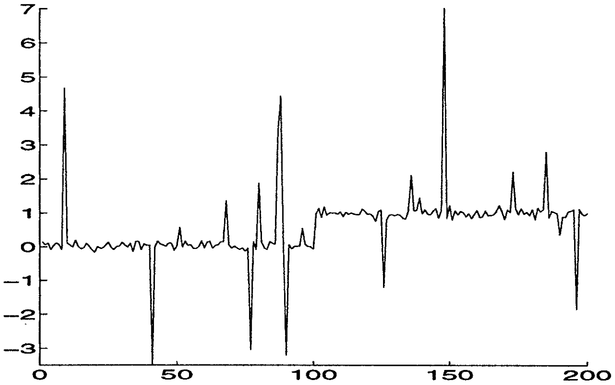


Figure 3.14. A unit step with heavy-tailed noise.

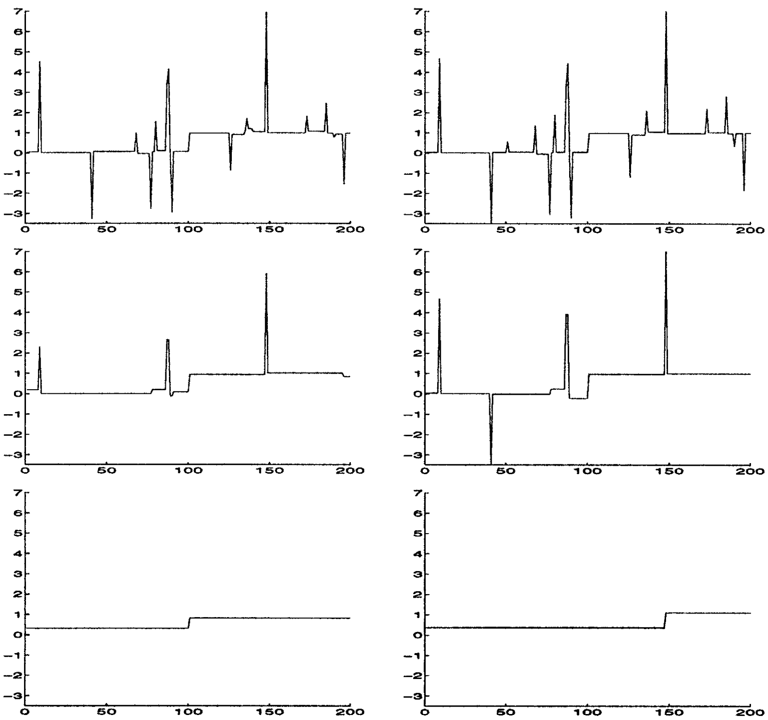
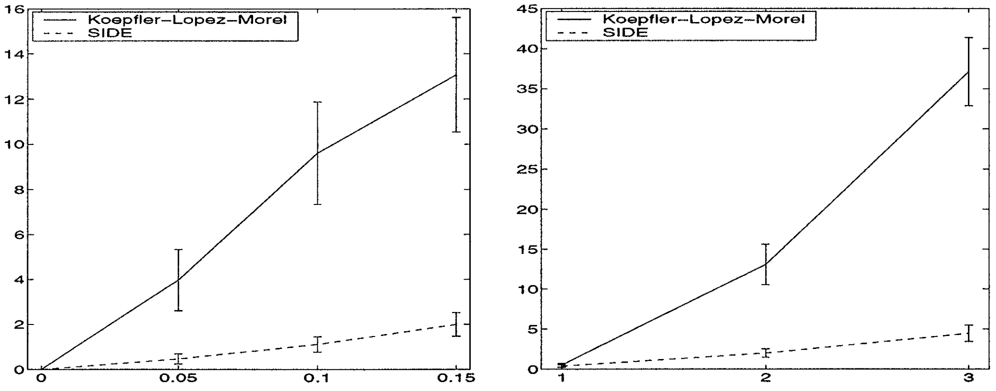


Figure 3.15. Scale spaces for the signal of Figure 3.14: SIDE (left) and Koepfler-Lopez-Morel (right). Top: 33 regions; middle: 11 regions; bottom: 2 regions.

We now compare the robustness of our algorithm to Koepfler, Lopez, and Morel's [36] region merging minimization of the Mumford-Shah functional [44]. For that purpose, we use Monte-Carlo simulations on a unit step signal corrupted by "heavy-tailed"



(a) (b)

Figure 3.16. Mean absolute errors for Monte-Carlo runs. (Koepfler-Lopez-Morel: solid line; SIDE: broken line.) The error bars are ±two standard deviations. (a) Different contamination probabilities (0, 0.05, 0.1, and 0.15); contaminating standard deviation is fixed at 2. (b) Contamination probability is fixed at 0.15; different contaminating standard deviations (1, 2, and 3).

noise which is, with high probability $1-\varepsilon$, normally distributed with $\sigma_1 = 0.1$, and, with low probability ε, normally distributed with a larger standard deviation σ_2. A typical sample path, for $\varepsilon = 0.1$ and $\sigma_2 = 2$, is shown in Figure 3.14. The SIDE and Koepfler-Lopez-Morel scale spaces for this signal are illustrated in Figure 3.15. During every Monte-Carlo trial, each algorithm was stopped when only two regions remained, and the resulting jump location was taken as the output. When $\sigma_2 = 2$, the mean absolute errors in locating the jump for $\varepsilon = 0$, $\varepsilon = 0.05$, $\varepsilon = 0.1$, and $\varepsilon = 0.15$ are shown in Figure 3.16(a) (the solid line is Koepfler-Lopez-Morel, the broken line is SIDE). The error bars are ±two standard deviations. Figure 3.16(b) shows the mean absolute errors for different standard deviations σ_2 of the contaminating Gaussian, when ε is fixed at 0.15.

As we anticipated in Chapter 2 and will further discuss in the next section, the quadratic term of the Mumford-Shah energy makes it non-robust to heavy-tailed noise, and the performance degrades considerably as the contamination probability and the variance of the contaminating Gaussian increase. Note that when $\sigma_2 = 3$ and $\varepsilon = 0.15$, using the Koepfler-Lopez-Morel algorithm is not significantly better than guessing the edge location as a random number between 1 and 200. At the same time, our method is very robust, even if the outlier probability is as high as 0.15.

Figure 3.17 shows the scale space generated by a Perona-Malik equation for the step signal with heavy-tailed noise depicted in Figure 3.14. As in Experiment 1, K was fixed at 90% of the histogram of the gradient, in accordance with Perona, Shiota, and Malik [50]. As before, its de-noising performance is good; however, it also introduces blurring and therefore its output does not immediately provide a segmentation. In order to get a good segmentation from this procedure, one needs to devise a stopping rule, so as to stop the evolution at a scale when noise spikes are diffused but the step

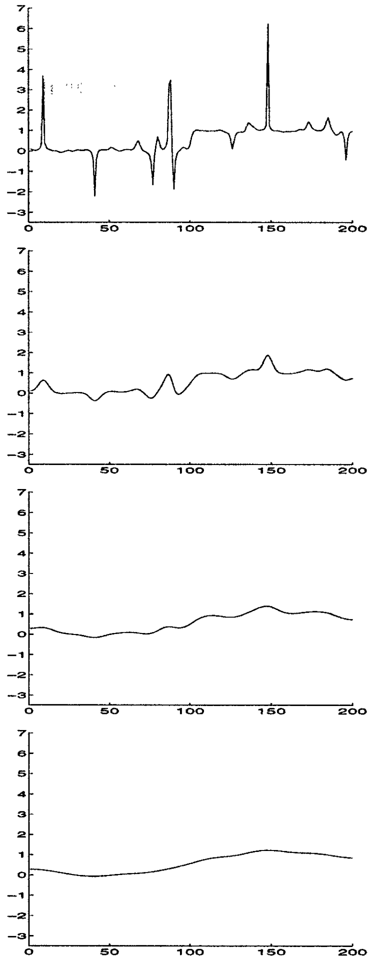


Figure 3.17. Scale space of a Perona-Malik equation with large K for the signal of Figure 3.14.

is not completely diffused (such as in the second plot of Figure 3.14). In addition, one needs to use an edge detector in order to extract the edge from the signal at that scale.

We again emphasize that neither SIDEs, nor the Koepfler-Lopez-Morel algorithm, nor any combination of the Perona-Malik equation with a stopping rule and an edge detector, are optimal for this simple 1-D problem, for which near-perfect results can be achieved in a computationally efficient manner by very simple procedures. The purpose of including this example is to provide statistical evidence for our claim of robustness of SIDEs. This becomes very important for complicated 2-D problems, such as the ones considered in the next example, where simple techniques no longer work.

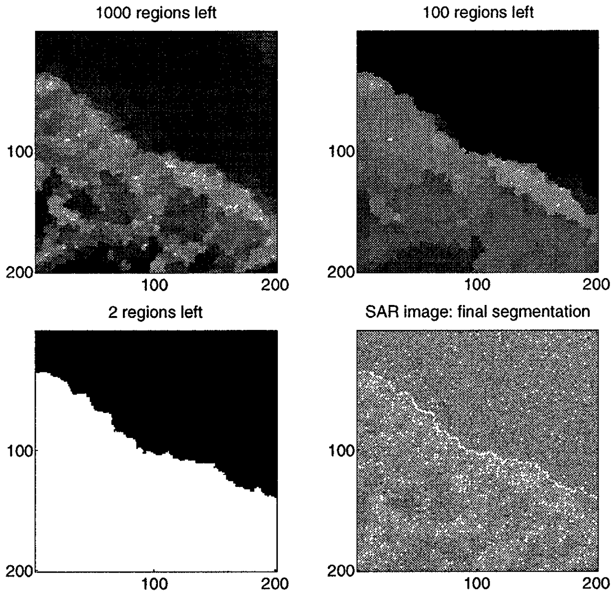


Figure 3.18. Scale space of a SIDE for the SAR image of trees and grass, and the final boundary superimposed on the initial image.

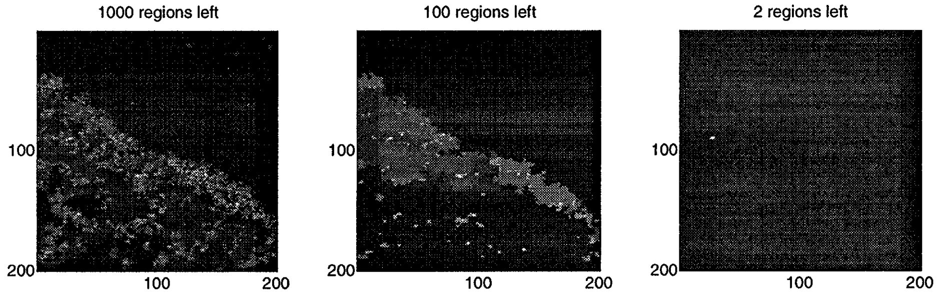
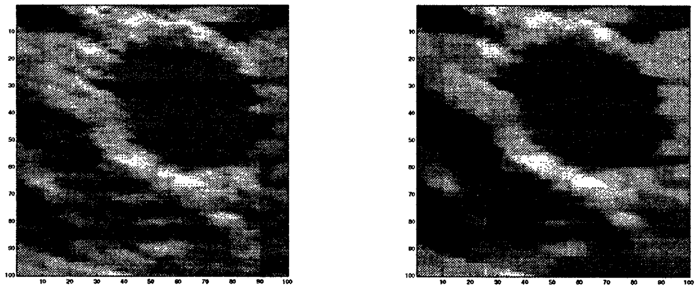


Figure 3.19. Segmentations of the SAR image via the region merging method of Koepfler, Lopez, and Morel.

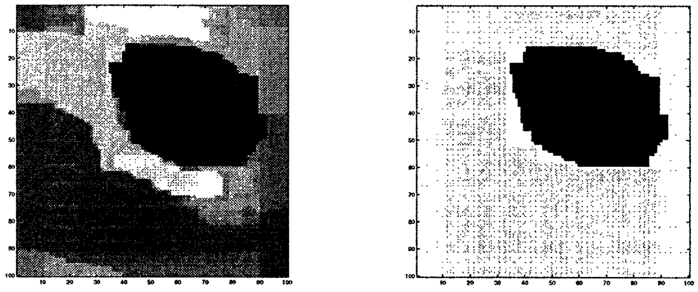
■ 3.5.4 Experiment 4: SIDE Evolutions in 2-D.

Both the sharpness of boundaries and robustness of SIDEs are also evident in the image experiments we have conducted. These properties are used to advantage in segmenting the SAR image of Figure 1.1 in which only two textures are present (forest and grass). The scale space is shown in Figure 3.18 (with the intensity values of each image scaled so as to take up the whole grayscale range), as well as the resulting boundary superimposed onto the original log-magnitude image. SAR imagery, such as the example shown here, are subject to the phenomenon known as speckle, which is present in any coherent imaging system and which leads to the large amplitude variations and noise evident in the original image. Consequently, the accurate segmentation of such imagery can be quite challenging and in particular cannot be accomplished using standard edge detection algorithms. For example, the scale space of the region merging algorithm of [36], as implemented in [20] and discussed above, is depicted in Figure 3.19. If evolved until two regions remain, it will find the boundary around a burst of noise. In contrast, the two-region segmentation displayed in Figure 3.18(d) is very accurate.

We note, that, as mentioned in Experiment 1, the SIDE evolutions require far less computation time than Perona-Malik-type evolutions. Since in 2-D a SIDE evolution is not a limiting form of a Perona-Malik evolution, the comparison is not quite as simple. However, in experiments that we have performed in which we have devised Perona-Malik evolutions that produce results as qualitatively similar to those in Figure 3.18 as possible, we have found that the resulting computational effort is roughly 130 times slower for this (201 × 201) image than our SIDE evolution. (As in 1-D, we used forward Euler discretization schemes [14] to implement both equations.)



(a) Ultrasound image of a thyroid. (b) 2000 regions.



(c) 170 regions. (d) 2 regions.

Figure 3.20. Scale space of a SIDE for the ultrasound image of a thyroid.

We close this section by presenting another segmentation example in 2-D. In the ultrasound image of a thyroid (Figure 3.20(a)), both speckle and blurring are evident. This degradation, inherent to ultrasound imaging, makes the problem of segmentation very difficult. Three images from the SIDE scale space are shown in Figure 3.20(b-d). The two-region segmentation (Figure 3.20(d)) is again very accurate.

We note that SIDEs have been used successfully in the context of another medical application in [19], namely segmentation of dermatoscopic images of skin lesions.

■ 3.6 Related Approaches.

■ 3.6.1 Mumford-Shah, Geman-Reynolds, and Zhu-Mumford.

The global energy (3.18) associated with SIDEs is similar to the first term of the image restoration model of D. Geman and Reynolds [21], as well as to Zhu and Mumford's potential function [75]. As we showed in the experimental section, this not only leads to sharp segmentations, but also allows our method to be more robust to heavy-tailed noise than algorithms which use quadratic energy terms. An important distinction from [21] and [75] is the interpretation of SIDEs as a region merging method, which leads to a much faster numerical implementation. It is also a more *universal* method of edge sharpening, since, unlike the algorithm of Geman and Reynolds, it does not require the knowledge of the blur model.

Since SIDEs are implemented via recursive region merging, it is instructive to compare them with other recursive region merging algorithms, such as Koepfler, Lopez, and Morel's [36] implementation of Mumford-Shah [44] segmentation, in which merging of neighboring regions occurs if it reduces the energy $\mathcal{E}_{MS}(\mathbf{u}) = (\mathbf{u}-\mathbf{u}^0)^T(\mathbf{u}-\mathbf{u}^0)+\lambda l$. (Here l is the total length of the boundaries, and λ is a scale parameter: a larger λ imposes greater penalty on boundaries, which results in a coarser segmentation $\mathbf{u}$.) The first term of this functional makes it non-robust to noise outliers. As we have seen, this term quadratically penalizes the difference between the initial image and its approximation $\mathbf{u}$, thereby causing very large outliers present in the original image $\mathbf{u}^0$ to re-appear in $\mathbf{u}$, even for large values of λ. This was clearly illustrated by the examples presented in the experimental section. The absence of such a term from the SIDE energy allows the evolution to diffuse strong bursts of noise, making it robust.

■ 3.6.2 Shock Filters and Total Variation.

Replacing the discrete vector $\mathbf{u}(t)$ with a function $u(t,x)$ of a continuous spatial variable x, and replacing first differences with derivatives in Equation (3.4), we see that, for $m_n = 1$, Equation (3.4) is a discretization of $u_t = \frac{\partial}{\partial x}[F(u_x)]$, where letter subscripts denote corresponding partial derivatives. Expanding the SIDE force function again as $F(v) = F_{id}(v) + C\ \mathrm{sgn}(v)$, we obtain:

$$u_t = C\frac{\partial}{\partial x}[\mathrm{sgn}(u_x)] + F'_{id}(u_x)u_{xx}. \tag{3.27}$$

The first of the RHS terms is the 1-D version of the gradient descent on total variation. It has very good noise removal properties, but, if used alone, it will ultimately blur the signal. If $F_{id}(v) = -\frac{1}{2}v|v|$, then the second term is equal to the RHS of one of the shock filters introduced by Osher and Rudin in [46]—namely, Equation (2.16) which we considered in Chapter 2. Discretizations of certain shock filters are excellent for edge enhancement, but, as we saw in Chapter 2, they cannot remove noise (and, in fact, some of them are unstable and noise-enhancing.) Thus, SIDEs combine the noise-suppressive properties of the total variation approach with the edge-sharpening features of shock filters. It should be noted, however, that (3.27) requires careful interpretation, because its RHS contains the signum function of u_x which itself may have both singularities and segments over which it is identically zero. In addition, this strange object is differentiated with respect to x. Thus, the interesting research issue arises of *defining* what one means by a solution to (3.27), in such a manner that the definition results in solutions relevant to the desired image processing applications. This complicated problem is avoided entirely with the introduction of SIDEs, since there one starts with a semi-discrete formulation, in which the issues of the existence and uniqueness of solutions are well understood. The SIDEs are thus a logical extension of Bouman and Sauer's approach of [6, 58] in which images are discrete matrices of numbers, rather than functions of two continuous variables. As described in Chapter 2, Bouman and Sauer proposed minimizing an energy functional consisting of two terms, one of which is the discrete counterpart of the total variation. Their method of quickly computing a local minimum of this non-differentiable functional involved merging pixels and thus anticipated SIDEs.

■ 3.7 Conclusion.

In this chapter we have presented a new approach to edge enhancement and segmentation, and demonstrated its successful application to signals and images with very high levels of noise, as well as to blurry signals. Our approach is based on a new class of evolution equations for the processing of imagery and signals which we have termed stabilized inverse diffusion equations or SIDEs. These evolutions, which have discontinuous right-hand sides, have conceptual and mathematical links to other evolution-based methods in signal and image processing, but they also have their own unique qualitative characteristics and properties. The next chapter is devoted to extensive analysis of a particular version of SIDEs.

Chapter 4

Probabilistic Analysis

■ 4.1 Introduction.

THE recent years have seen a great number of exciting developments in the field of nonlinear diffusion filtering of images. As summarized in Chapter 2 and Section 3.6, many theories have been proposed that result in edge-preserving scale spaces possessing various interesting properties. One striking feature unifying many of these frameworks–including the one introduced in the previous chapter–is that they are deterministic. Usually, one starts with a set of "common-sense" principles which an image smoothing operation should satisfy. Examples of these are the axioms in [1] and the observation in [49] that, in order to achieve edge preservation, very little smoothing should be done at points with high gradient. From these principles, a nonlinear scale space is derived, and then it is analyzed–again, deterministically. Note, however, that since the objective of these techniques is usually restoration or segmentation of images in the presence of noise, a natural question to ask would be:

> Do nonlinear diffusion techniques solve standard estimation or detection problems? (*)

An affirmative answer would help us understand which technique is suited best for a particular application, and aid in designing new algorithms. It would also put the tools of the classical detection and estimation theory at our disposal for the analysis of these techniques, making it easier to tackle an even more crucial question:

> Given a probabilistic noise model, can one characterize the performance of the nonlinear diffusion techniques? (**)

Attempts to address these issues in the literature have remained scarce–most likely, because the complex nature of the nonlinear partial differential equations (PDEs) considered and of the images of interest make this analysis prohibitively complicated. Most notable exceptions are [63,75] which establish qualitative relations between the Perona-Malik equation [49] and gradient descent procedures for estimating random fields modeled by Gibbs distributions. Bayesian ideas are combined in [76] with snakes and region growing for image segmentation. In [5], concepts from robust statistics are used to

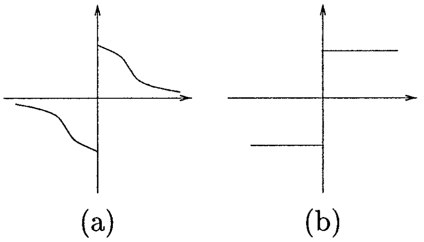


Figure 4.1. Functions F from the right-hand side of the SIDE: (a) generic form; (b) the signum function.

modify the Perona-Malik equation. In [38], a connection between random walks and diffusions is used to obtain a new evolution equation.

The goal of this chapter is to move forward the discussion of questions (*) and (**). We consider a very simple nonlinear diffusion (a variant of those introduced in the previous chapter) which provides a multiscale sequence of segmentations of its initial condition. In Sections 4.3 and 4.4, we apply our algorithm to 1-D signals, and describe edge detection problems which are solved optimally by this diffusion. These are binary classification problems: each sample has to be classified as coming from one of two classes, subject to the constraint on the number of "edges"—i.e., changes between the two classes. One of these problems turns out to be the minimization of a special case of the 1-D Mumford-Shah functional [44]. We describe an efficient implementation of the 1-D diffusion, requiring $O(N \log N)$ computations in the worst case, where N is the size of the input signal. In Section 4.5, we point out that the same 1-D problem can also be solved via dynamic programming and via linear programming, but that our method has certain advantages over both. To analyze the performance (Section 4.6), we simplify even further, by considering signals with only one change in mean. Our performance measure is the accuracy in locating the change. More precisely, the probability of events of the form "the detected change location is more than p samples away from the actual one" is analyzed. We show that the asymptotic probabilities of these events can be obtained directly from the classical change detection paper by Hinkley [28]. We also derive non-asymptotic lower bounds on these probabilities. The robustness of the algorithm—which is experimentally confirmed both in this chapter and in Chapter 3—is analyzed theoretically by showing the optimality with respect to a certain H_∞-like criterion. In Section 4.7, we treat segmentation of 2-D images.

■ 4.2 Background and Notation.

In this chapter, we consider a special case of SIDEs (3.11) in 1-D, which results if one drops the "monotonically decreasing" requirement on F, and takes $F(v) = \text{sgn}(v)$ instead (see Figure 4.1, (b)). Specifically, we are interested in the evolution of the following equation:

$$\begin{aligned} \dot{u}_1 &= \frac{\operatorname{sgn}(u_2 - u_1)}{m_1}, \quad \dot{u}_N = \frac{\operatorname{sgn}(u_{N-1} - u_N)}{m_N}, \\ \dot{u}_n &= \frac{1}{m_n}(\operatorname{sgn}(u_{n+1} - u_n) - \operatorname{sgn}(u_n - u_{n-1})), \\ & \text{for } n = 2, \ldots, N-1, \end{aligned} \tag{4.1}$$

with the initial condition

$$\mathbf{u}(0) = \mathbf{u}^0, \tag{4.2}$$

where, as we explain in Section 4.4, $\mathbf{u}^0$ is either the signal to be processed or a sequence of logarithms of pointwise likelihood ratios. As in the previous chapter, N stands for the number of samples in the signals under consideration. Boldface letters denote these signals, whose entries are always denoted by the same letter with subscripts 1 through N: $\mathbf{u} = (u_1, \ldots u_N)^T$. Initially the finest segmentation is assumed: each pixel is a separate region, i.e. $m_n = 1$, for $n = 1, \ldots, N$. As soon as u_i becomes equal to u_{i+1}, these values stay equal forever, and their equations are replaced with

$$\dot{u}_i = \dot{u}_{i+1} = \frac{(\operatorname{sgn}(u_{i+2} - u_{i+1}) - \operatorname{sgn}(u_i - u_{i-1}))}{m_i + m_{i+1}}, \tag{4.3}$$

which is Equation (3.10). The rest of the present chapter deals with the particular SIDE (4.1,4.2,4.3).

We apply the SIDE (4.1,4.2) to binary classification problems. Given an observation $\mathbf{y}$, the goal is to label each sample as coming from one of two classes, i.e. to produce a binary signal $\mathbf{h}$ whose entries are zeros and ones. We call any such binary signal $\mathbf{h}$ an *hypothesis.*

Definition 4.1. *We denote the set of all N-dimensional binary signals by* $\{0,1\}^N$. *A member* $\mathbf{h}$ *of this set is called an* **hypothesis** *with the following interpretation. Specifically, if* $h_i \neq h_{i+1}$, *we say that an* **edge** *(or, equivalently, a* **change***) is hypothesized at the location i, and we call sgn*$(h_{i+1} - h_i)$ **the sign of the edge**. *We say that an edge is directed* **upward (downward)** *if* $h_{i+1} > h_i$ $(h_{i+1} < h_i)$. ■

Definition 4.2. *A* **statistic** *is a function* $\phi : \mathbb{R}^N \times \{0,1\}^N \to \mathbb{R}$. *The* **optimal** *hypothesis* $\mathbf{h}^*(\mathbf{u})$ *for a signal* $\mathbf{u} \in \mathbb{R}^N$ *with respect to* ϕ *is*

$$\mathbf{h}^*(\mathbf{u}) \stackrel{\text{def}}{=} \arg \max_{\mathbf{h} \in \{0,1\}^N} \phi(\mathbf{u}, \mathbf{h}).$$

The best hypothesis among those whose number of edges does not exceed some constant ν *is*

$$\mathbf{h}^*_{\leq \nu}(\mathbf{u}) \stackrel{\text{def}}{=} \arg \max_{\mathbf{h} \in \{0,1\}^N,\ \mathbf{h} \text{ has } \nu \text{ or fewer edges}} \phi(\mathbf{u}, \mathbf{h}). \quad ■$$

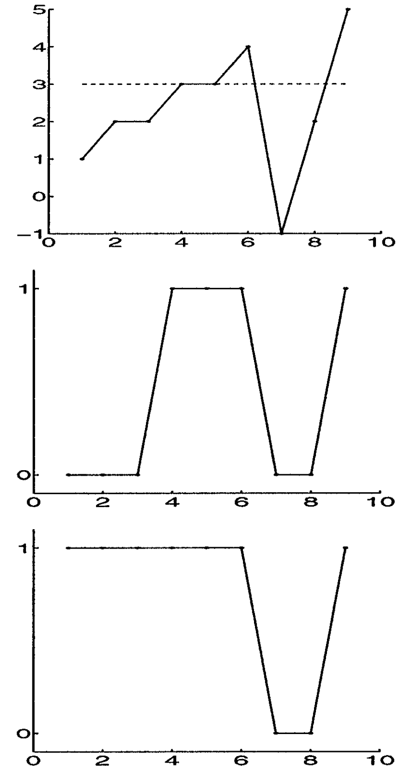


Figure 4.2. Illustrations of Definitions 4.1 and 4.3: a sequence with three α-crossings, where α=3 (top); the hypothesis generated by the three α-crossings (middle); the hypothesis generated by the two rightmost α-crossings (bottom).

Note that an hypothesis is uniquely defined by the set of its edges and the sign of one of the edges. Therefore, binary classification problems can also be viewed as edge detection problems. For the problems considered in this chapter, the optimal edge locations will typically be level crossings of some signal.

Definition 4.3. *A signal* $\mathbf{u}$ *is said to have an* **α-crossing** *at the location* i *if* $(u_i - \alpha)(u_j - \alpha) < 0$, *where* $j = min\{n\colon n > i, u_n \neq \alpha\}$. *(In other words,* u_j *is the first sample to the right of* i *which is not equal to* α.*) We call* $sgn(\alpha - u_i)$ **the sign of the α-crossing**, *and say that the* α*-crossing is directed* **upward (downward)** *if* $u_i < \alpha$ $(u_i > \alpha)$. *We define the hypothesis* **generated** *by a set of* α*-crossings* $\{g_1, \dots, g_\nu\}$ *of* $\mathbf{u}$ *as the hypothesis whose edges are at* $g_1, \dots, g_\nu$ *and for which the sign of each edge is the same as the sign of the corresponding* α*-crossing.* ■

To illustrate Definitions 4.1, 4.2, and 4.3, let us consider an example.

Example 4.1. *Illustration of the definitions of edges,* α*-crossings, and statistics.*

Suppose

$$\mathbf{u} = (1, 2, 2, 3, 3, 4, -1, 2, 5)^T$$

(see top of Figure 4.2), and $\alpha = 3$. Then $\mathbf{u}$ has three α-crossings, at locations $g_1 = 3$, $g_2 = 6$, and $g_3 = 8$. The second one is directed downward, and the other two are directed upward. The hypothesis $\mathbf{h}_1$ generated by these three α-crossings must therefore have upward edges at 3 and 8 and a downward edge at 6:

$$\mathbf{h}_1 = (0,0,0,1,1,1,0,0,1)^T,$$

as depicted in the middle plot of Figure 4.2. The hypothesis $\mathbf{h}_2$ generated by the α-crossings g_2 and g_3 will only have a downward edge at 6 and an upward edge at 8:

$$\mathbf{h}_2 = (1,1,1,1,1,1,0,0,1)^T,$$

as shown in the last plot of Figure 4.2. If we define a statistic ϕ by

$$\begin{aligned} \phi(\mathbf{u},\mathbf{h}) &= \mathbf{h}^T(\mathbf{u}-\mathbf{a}), \\ \text{where } \mathbf{a} &= (3,\ldots,3)^T \in \mathbb{R}^9, \end{aligned}$$

then we have:

$$\begin{aligned} \phi(\mathbf{u},\mathbf{h}_1) &= 3, \\ \phi(\mathbf{u},\mathbf{h}_2) &= -1. \quad \blacksquare \end{aligned}$$

■ 4.3 SIDE as an Optimizer of a Statistic.

The usefulness of the SIDE (4.1,4.2) in solving edge detection problems comes from its ability to maximize certain statistics. One of the properties of such a statistic $\phi(\mathbf{u},\mathbf{h})$ is that the optimal hypothesis $\mathbf{h}^*(\mathbf{u})$ is generated by the set of all α-crossings of $\mathbf{u}$, for some number α. It is shown below in Proposition 4.1 that every edge of $\mathbf{h}^*_{\le\nu}(\mathbf{u})$ also occurs at an α-crossing of $\mathbf{u}$. Furthermore, Proposition 4.2 describes how to find these edges using the SIDE (4.1): it says that the α-crossings of the solution $\mathbf{u}(t)$ to the SIDE (4.1) then generate the constrained optimal hypothesis $\mathbf{h}^*_{\le\nu}(\mathbf{u}^0)$, where $\mathbf{u}^0$ is the initial data, and ν is the number of α-crossings of $\mathbf{u}(t)$. We note that when we talk about α-crossings of $\mathbf{u}$ in Propositions 4.1 and 4.2, we will allow α to be a function of $\frac{1}{N}\sum_{i=1}^N u_i$. This is necessary to cover some important examples considered later in Section 4.4. On the other hand, $\frac{1}{N}\sum_{i=1}^N u_i(t)$ is constant throughout the evolution of the SIDE, as easily verified by summing up the equations (4.1). Thus, $\alpha(\frac{1}{N}\sum_{i=1}^N u_i)$ will also stay constant during the evolution of the SIDE. We therefore will simply refer to α, dropping its argument to avoid notational clutter. It will, however, be our implicit assumption throughout the remainder of this chapter that whenever we mention α-crossings of a signal $\mathbf{u}$, α is allowed to be a function of $\frac{1}{N}\sum_{i=1}^N u_i$.

Proposition 4.1. *Let* $\alpha : I\!R \to I\!R$ *be a scalar function, and let* $\mathbf{a} : I\!R \to I\!R^N$ *be the vector function whose every entry is* α: $\mathbf{a}(x) = (\alpha(x), \ldots, \alpha(x))^T$ *for any* $x \in I\!R$. *Define the following statistic:*

$$\phi(\mathbf{u}, \mathbf{h}) = \mathbf{h}^T \left(\mathbf{u} - \mathbf{a}\left(\frac{1}{N} \sum_{i=1}^{N} u_i \right) \right). \tag{4.4}$$

Then every edge of $\mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$ *occurs at an* α*-crossing of* $\mathbf{u}^0$, *for any* $\mathbf{u}^0 \in I\!R^N$.

Proof is in Appendix B. ■

Proposition 4.2. *Suppose that* ϕ *is the statistic described in Proposition 4.1. Fix the initial condition* $\mathbf{u}^0$ *of the SIDE (4.1), and let* $\mathbf{u}(t)$ *be the corresponding solution. Then* $\alpha(\frac{1}{N}\sum_{i=1}^{N} u_i(t))$ *is constant during the evolution of the SIDE, as verified by summing up the equations (4.1). Let* $\nu_\alpha(t)$ *be the number of* α*-crossings of* $\mathbf{u}(t)$. *Then, for any time instant* $t_f > 0$,

$$\mathbf{h}^*_{\leq\nu_\alpha(t_f)}(\mathbf{u}^0) = \mathbf{h}^*(\mathbf{u}(t_f)). \qquad \blacksquare$$

The proof is in Appendix B. We note that Proposition 1 of [51] is a different formulation of the same result: in [51], we explicitly listed the properties of ϕ which are used in the proof. The equivalence of the two formulations is also shown in Appendix B.

This proposition says that, if the SIDE is evolved until $\nu_\alpha(t)$ α-crossings remain, then these α-crossings are the optimal edges, where "optimality" means maximizing the statistic $\phi(\mathbf{u}^0, \mathbf{h})$ subject to the constraint that the hypothesis have $\nu_\alpha(t)$ or fewer edges. It is verified in the next subsection that $\nu_\alpha(t)$ is a non-increasing function of time, with $\nu_\alpha(\infty) = 0$. Unfortunately, $\nu_\alpha(t)$ is not guaranteed to assume every integer value between $\nu_\alpha(0)$ and 0: during the evolution of the SIDE, α-crossings can disappear in pairs. We will show in the next subsection, however, that no more than two α-crossings can disappear at the same time. We will also show that, even if for some integer $\nu < \nu_\alpha(0)$ there is no t such that $\nu_\alpha(t) = \nu$ (i.e. $\nu_\alpha(t)$ goes from $\nu + 1$ directly to $\nu - 1$), we can still easily find $\mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$ using the set of α-crossings of the solution $\mathbf{u}(t)$ to the SIDE at the time t when $\nu_\alpha(t) = \nu + 1$. If the desired number of edges is greater than or equal to the initial number of α-crossings, $\nu \geq \nu_\alpha(0)$, then, from the definitions of $\mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$ and $\mathbf{h}^*(\mathbf{u}^0)$, we immediately have:

Proposition 4.3. *Suppose that* ϕ *is the statistic described in Proposition 4.1. If* $\nu \geq \nu_\alpha(0)$, *then*

$$\mathbf{h}^*_{\leq\nu}(\mathbf{u}^0) = \mathbf{h}^*(\mathbf{u}^0). \qquad \blacksquare$$

In the remainder of the chapter, we assume that $\nu < \nu_\alpha(0)$.

In Section 4.4, we will give examples of detection problems whose solution is equivalent to maximizing the statistic ϕ. We will therefore be able to utilize the SIDE for optimally solving these problems. Before we do that, however, we describe how to efficiently implement the SIDE.

■ 4.3.1 Implementation of the SIDE Via a Region Merging Algorithm.

We now show how to solve efficiently the optimization problem treated in Proposition 4.2 of the previous subsection, using the SIDE (4.1).

Problem 4.1. *Define a statistic*

$$\phi(\mathbf{u}, \mathbf{h}) = \mathbf{h}^T \left(\mathbf{u} - \mathbf{a}\left(\frac{1}{N} \sum_{i=1}^{N} u_i \right) \right),$$

where $\mathbf{a}(x) = (\alpha(x), \ldots, \alpha(x))^T \in I\!R^N$*, and* α *is a real-valued function of a real argument. Given an integer* ν *and a signal* $\mathbf{u}^0$*, we are interested in finding the best hypothesis* $\mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$ *among all the hypotheses with* ν *or fewer edges, where "the best" means the one maximizing* $\phi(\mathbf{u}^0, \cdot)$*.* ■

Proposition 4.2 relates $\mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$ to the solution $\mathbf{u}(t)$ of the SIDE whose initial data is $\mathbf{u}^0$. Namely, it says that if ν is the number of α-crossings of $\mathbf{u}(t)$[1], then these α-crossings generate the hypothesis $\mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$. It is, however, not guaranteed that for every integer ν there is a time instant t when the solution $\mathbf{u}(t)$ has exactly ν α-crossings. Therefore, in order to compute the solution to Problem 4.1, we need to deal with two issues:

(A) How to calculate the locations of the α-crossings of $\mathbf{u}(t)$?

(B) How to compute $\mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$ for *every* integer ν?

We first consider issue A. In order to find the α-crossings of the solution to the SIDE, one can certainly use a finite difference scheme to numerically integrate the equation. There is, however, a much faster way, which exploits the special structure of the equation. It turns out that, during the evolution of the SIDE, α-crossings cannot be created or shifted: they can only be erased. We therefore only need to compute the order in which they disappear. We now make these statements precise.

Lemma 4.1. *Suppose that at time* t_0*, the solution* $\mathbf{u}(t_0)$ *to the SIDE has no* α*-crossing at the location* i*. Then* $\mathbf{u}(t)$ *has no* α*-crossing at* i*, either, for all* $t \geq t_0$*.*

Example 4.2 (Example 4.1, continued). *Illustration of Lemma 4.1.*

We illustrate this Lemma by evolving the SIDE (4.1) for the initial condition

$$\mathbf{u}^0 = (1, 2, 2, 3, 3, 4, -1, 2, 5)^T$$

(top of Figure 4.2) and $\alpha = 3$. The values of the solution for several time instants are recorded in Table 1.

---

[1] Just as in the previous subsection, we abuse notation by dropping the argument $\frac{1}{N}\sum_{i=1}^{N} u_i(t)$ of α.

Table 1: Evolution of the SIDE.

t	$\mathbf{u}^T(t)$
$t = 0$	$(1, 2, 2, 3, 3, 4, -1, 2, 5)$
$t = \frac{1}{2}$	$(1\frac{1}{2}, 2, 2, 3, 3, 3, 0, 2, 4\frac{1}{2})$
$t = 1$	$(2, 2, 2, 2\frac{2}{3}, 2\frac{2}{3}, 2\frac{2}{3}, 1, 2, 4)$
$t = 1\frac{1}{2}$	$(2\frac{1}{6}, 2\frac{1}{6}, 2\frac{1}{6}, 2\frac{1}{3}, 2\frac{1}{3}, 2\frac{1}{3}, 2, 2, 3\frac{1}{2})$
$t = 1\frac{2}{3}$	$(2\frac{2}{9}, 2\frac{2}{9}, 2\frac{2}{9}, 2\frac{2}{9}, 2\frac{2}{9}, 2\frac{2}{9}, 2\frac{1}{6}, 2\frac{1}{6}, 3\frac{2}{3})$
$t = 2$	$(2\frac{1}{4}, 2\frac{1}{4}, 2\frac{1}{4}, 2\frac{1}{4}, 2\frac{1}{4}, 2\frac{1}{4}, 2\frac{1}{4}, 2\frac{1}{4}, 3\frac{1}{3})$
$t = 2\frac{2}{3}$	$(2\frac{1}{3}, 2\frac{1}{3}, 2\frac{1}{3}, 2\frac{1}{3}, 2\frac{1}{3}, 2\frac{1}{3}, 2\frac{1}{3}, 2\frac{1}{3}, 2\frac{2}{3})$

At the beginning, $\mathbf{u}(t)$ has three α-crossings, at locations 3, 6, and 8. At time $t = \frac{1}{2}$, the first two α-crossings disappear. Between $t = 2$ and $t = 2\frac{2}{3}$, the α-crossing located at 8 also disappears. ■

In order to prove Lemma 4.1, we need to re-define a *region* (i, j) of a signal $\mathbf{u}$ as the set of all samples $\{u_i, \dots, u_j\}$ either between two consecutive α-crossings $i - 1$ and j of $\mathbf{u}$, or to the left of the leftmost α-crossing j (in which case $i = 1$), or to the right of the rightmost α-crossing $i - 1$ (in which case $j = N$)[2]. For example, referring to Table 1, $\mathbf{u}(0)$ has four regions: $(1, 3)$, $(4, 6)$, $(7, 8)$, and $(9, 9)$. Similarly, $\mathbf{u}(2)$ has two regions: $(1, 8)$ and $(9, 9)$. Note that a sample $u_k(t)$ belonging to a region (i, j) can only cross the level α if all other samples in its region do so at the same time:

Lemma 4.2. *Let* (i, j) *be a region of* $\mathbf{u}(t_0)$*, where* $t_0 \geq 0$*, and let the intensity values inside this region be above* α*:* $u_i(t_0) > \alpha, \dots, u_j(t_0) > \alpha$*. Let* t_1 *be the first time instant after* t_0 *at which one of the values inside the region, say* $u_k(t_1)$*, becomes equal to* α*:*

$$t_1 = \inf\{t > t_0 : \ \exists k, \ i \leq k \leq j, u_k(t) = \alpha\}.$$

Then

$$u_i(t_1) = u_{i+1}(t_1) = \dots u_j(t_1).$$

Proof. Notice that, according to Equations (4.1,4.3), $u_k(t)$ can be decreasing only if it is a local maximum (i.e., if $u_k(t) \geq u_{k\pm1}(t)$). Thus, at time t_1, we have that

- $u_k(t_1) = \alpha$ is a local maximum;
- all other values inside the region (i, j) are $\geq \alpha$.

Consequently, it must be that the values at all the samples inside the region (i, j) are equal to α. ■

Proof of Lemma 4.1. A proof similar to the one above applies to the variant of Lemma 4.2 when $u_i(t_0), \dots, u_j(t_0)$ are less than α. Thus, the value $u_k(t)$ at any location k can

---

[2] Note that this definition is somewhat different from the one in Chapter 3, where all pixels in a region had the same value.

cross the level α only when its whole region does so. This means that the evolution of the SIDE cannot create or shift α-crossings; it can only remove them. ∎

We now show how to calculate the order in which the regions disappear. It turns out that this ordering depends on how the removal of a region influences the statistic $\phi(\mathbf{u}(t), \mathbf{h})$. Define the *energy* E_{ij} of the region (i, j) by

$$\begin{aligned} E_{ij}(t) &= \frac{1}{\rho_{ij}} \left| \sum_{n=i}^{j} (u_n(t) - \alpha) \right|, \qquad (4.5) \\ \rho_{ij} &= \begin{cases} 1 & \text{if } i = 1 \text{ or } j = N \\ 2 & \text{otherwise.} \end{cases} \end{aligned}$$

Note that the energy measures the contribution of the region (i, j) to $\phi(\mathbf{u}(t), \mathbf{h})$. Summing up the equations (4.1) from $n = i$ to $n = j$, we see that, for every region (i, j), $\dot{E}_{ij}(t) = -1$. A region (i, j) is erased when the values of all its samples reach α—i.e., when the energy $E_{ij}(t)$ becomes equal zero. Since all the energies are diminished with the same speed, it follows that the first region to disappear will be the one for which $E_{ij}(0)$ is the smallest. Applying this reasoning recursively, we then remove the region with the next smallest energy, etc., obtaining the following algorithm to compute the α-crossings of $\mathbf{u}(t)$.

1. **Initialize.** Let A be the set of all α-crossings of $\mathbf{u}^0$, ordered from left to right, and let $\bar{\nu} = \nu_\alpha(0)$ be the total number of α-crossings.

2. **Compute the energies.** Denote the elements of the set A by $g_1, \ldots, g_{\bar{\nu}}$, and form $\bar{\nu} + 1$ regions: $(1, g_1), (g_1 + 1, g_2), \ldots, (g_{\bar{\nu}} + 1, N)$. For each region (i, j), compute its energy E_{ij}, as defined by (4.5).

3. **Remove the region with minimal energy.** Let (i_m, j_m) be the region for which E_{ij} is the smallest (if there are several regions with the smallest energy, choose any one). Merge the region (i_m, j_m) with its neighbors by re-defining A and $\bar{\nu}$ via

$$A \leftarrow A \backslash \{i_m, j_m\}, \quad \bar{\nu} \leftarrow \text{ the size of the new } A.$$

4. **Iterate.** If $\bar{\nu}$ is greater than the desired number of edges, go to step 2. ∎

To illustrate the algorithm, we apply it to the signal of Example 4.1 (top of Figure 4.2).

Example 4.3 (Examples 4.1 and 4.2, continued). *Illustration of the region merging algorithm.*

We again consider

$$\mathbf{u}^0 = (1, 2, 2, 3, 3, 4, -1, 2, 5)^T.$$

Iteration 1. There are three α-crossings, and four regions: $(1, 3)$, $(4, 6)$, $(7, 8)$, and $(9, 9)$, with the energies $E_{13} = 4$, $E_{46} = 0.5$, $E_{78} = 2.5$, and $E_{99} = 2$, respectively. The region $(4,6)$ has the smallest energy, and therefore it is removed first, by merging it with its two neighbors to form the new region $(1,8)$.

Iteration 2. There are now two regions, $(1,8)$ and $(9,9)$, with the energies $E_{1,8} = 8$ and $E_{99} = 2$, respectively. They are merged, to form one region $(1,9)$. Note that the order in which the regions disappear is in agreement with Table 1. ■

We now show that the algorithm is fast. The initialization steps 1 and 2 take $O(N)$ time. Step 3 merges either two regions (if $i_m = 1$ or $j_m = N$) or three regions (otherwise). The energy of the new region is essentially the sum of the energies of its constituent regions, and therefore the recomputation of energies after a merging takes $O(1)$ time. If a binary heap [11] is used to store the energies, the size of the heap at every iteration of the algorithm will be equal to the current number of distinct regions, which is $\bar{\nu} + 1$, where $\bar{\nu}$ is the current number of α-crossings. Therefore, finding the minimal element of the heap (step 3) at each iteration will take $O(\log \bar{\nu} + 1)$ time, which means that the algorithm will run in $O(\sum_{\bar{\nu}=\nu+1}^{\nu_\alpha(0)} \log \bar{\nu} + N)$ time. The worst case is when $\nu = 1$ and $\nu_\alpha(0) = N - 1$. Then the computational complexity is $O(N \log N)$. However, if the desired number of edges ν is comparable with the initial number $\nu_\alpha(0)$ (which can happen in low-noise scenarios), the complexity is $O(N)$.

We still have to address Question (B) which we posed at the beginning of this subsection, namely, how to find $\mathbf{h}^*_{\le\nu}(\mathbf{u}^0)$ for *every* integer ν. If there is a time instant t at which $\mathbf{u}(t)$ has exactly ν α-crossings, then, according to Proposition 4.2, these α-crossings generate the hypothesis $\mathbf{h}^*_{\le\nu}(\mathbf{u}^0)$, which means that Problem 4.1 is solved. The scenario which we need to consider now is when there is no such time t at which $\mathbf{u}(t)$ has exactly ν α-crossings. As we showed above, computing the locations of α-crossings of $\mathbf{u}(t)$ involves removing regions one at a time (Step 3 of the algorithm). Thus, at most two α-crossings can disappear at the same time. Therefore, if $\mathbf{u}(t)$ never has ν α-crossings, it must go from $\nu + 1$ α-crossings directly to $\nu - 1$. It turns out that, in this case, one can still compute $\mathbf{h}^*_{\le\nu}(\mathbf{u}^0)$ by running the algorithm above until $\nu - 1$ α-crossings remain, and then doing post-processing whose computational complexity is $O(N)$. Specifically, the following proposition holds.

Proposition 4.4. *Suppose that there is a time instant t during the evolution of the SIDE such that $\mathbf{u}(t^-)$ has $\nu + 1$ α-crossings, at locations $g_1, \dots, g_{\nu+1}$. The hypothesis generated by these α-crossings is $\mathbf{h}^*_{\le\nu+1}(\mathbf{u}^0)$. Suppose further that the region $(g_k + 1, g_{k+1})$ disappears at time t, so that $\mathbf{u}(t^+)$ has $\nu - 1$ α-crossings, at locations $g_1, \dots, g_{k-1}, g_{k+2}, \dots, g_{\nu+1}$. The hypothesis generated by these α-crossings is $\mathbf{h}^*_{\le\nu-1}(\mathbf{u}^0)$. Then one of the following four possibilities must happen.*

(i) $\mathbf{h}^*_{\le\nu}(\mathbf{u}^0) = \mathbf{h}^*_{\le\nu-1}(\mathbf{u}^0)$.

(ii) $\mathbf{h}^*_{\le\nu}(\mathbf{u}^0)$ *has edges at the locations* $g_2, \dots, g_{\nu+1}$.

(iii) $\mathbf{h}^*_{\le\nu}(\mathbf{u}^0)$ *has edges at the locations* $g_1, \dots, g_\nu$.

(iv) $\mathbf{h}^*_{\le\nu}(\mathbf{u}^0)$ *has edges at the locations* $g_1, \dots, g_{k-1}, g_{k+2}, \dots, g_{\nu+1}$*, as well as one edge at the location which is an element of the set* $\{1, 2, \dots, g_1 - 1, g_{\nu+1} + 1, \dots, N - 1\}$.

Thus, finding $\mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$ *is achieved by running the SIDE and doing post-processing of complexity* $O(N)$. ■

Proposition 4.4 is the recipe for obtaining the optimal hypothesis $\mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$ from the $\nu+1$ α-crossings of $\mathbf{u}(t^-)$. It says that either ν or $\nu-1$ of these α-crossings coincide with the edges of $\mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$. Cases (ii) and (iii) describe the only two subsets consisting of ν α-crossings which can generate $\mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$. If only $\nu-1$ α-crossings of $\mathbf{u}(t^-)$ coincide with the edges of $\mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$, then either there are no other edges (Case (i)), or the remaining edge is easily found in linear time (Case (iv)).

We note that a slight correction to what was reported in [51] is in order: although the statement that the complexity of the post-processing is $O(N)$ is correct, the specific post-processing procedure given there is somewhat different from the one outlined in Proposition 4.4 above, and therefore, it may be *incorrect* for some data sequences.

As one can infer from the statement of this proposition, its proof is rather technical and amounts to analyzing various scenarios of the disappearance of the α-crossings. This proposition is a direct corollary of Proposition 4.1 and the following lemma.

Lemma 4.3. *As in Proposition 4.4, let* t *be the time instant when the solution of the SIDE goes from* $\nu+1$ *α-crossings to* $\nu-1$. *Let* $\mathbf{h}=\mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$, *and suppose that it is generated by* ν *α-crossings of* $\mathbf{u}^0$: $g_1,\dots,g_\nu$. *(Note that this notation is* different *from the notation of Proposition 4.4.) Then at least* $\nu-1$ *elements of the set* $\{g_1,\dots,g_\nu\}$ *are also α-crossings of* $\mathbf{u}(t^-)$, *with possible exception of either* g_1 *or* g_ν. *Furthermore, if exactly* $\nu-1$ *elements of the set* $\{g_1,\dots,g_\nu\}$ *are α-crossings of* $\mathbf{u}(t^-)$, *they are also α-crossings of* $\mathbf{u}(t^+)$,

Proof is in Appendix B. ■

4.4 Detection Problems Optimally Solved by the SIDE.

We now give examples of detection problems whose solution is equivalent to maximizing the statistic ϕ of Proposition 4.2. These problems can therefore be optimally solved by the SIDE.

4.4.1 Two Distributions with Known Parameters.

Let $\mathbf{y}$ be an observation of a sequence of N independent random variables. Suppose that each random variable has probability density function (pdf) either $f(y,\theta_0)$ or $f(y,\theta_1)$, where θ_0 and θ_1 are known. It is also known that the number of changes between the two pdf's does not exceed ν; however, it is not known where these changes occur.

To obtain the maximum likelihood hypothesis [66], one has to maximize the log likelihood function

$$\sum_{i:h_i=1}\log f(y_i,\theta_1)+\sum_{i:h_i=0}\log f(y_i,\theta_0),$$

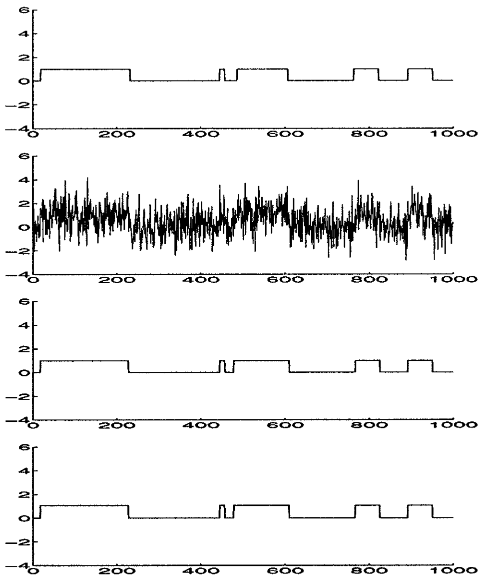


Figure 4.3. Edge detection for a binary signal in Gaussian noise.

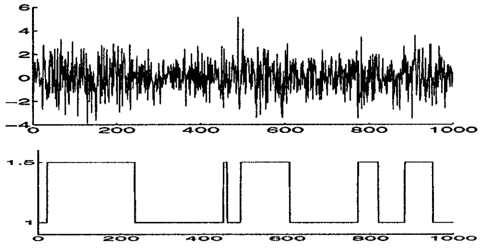


Figure 4.4. Detection of changes in variance of Gaussian noise.

where the hypothesis $\mathbf{h}$ is such that the sample y_i is hypothesized to be from the pdf $f(y, \theta_{h_i})$. Note that by defining a signal consisting of pointwise log-likelihoods,

$$u_i^0 = \log f(y_i, \theta_1) - \log f(y_i, \theta_0), \tag{4.6}$$

we see that the log-likelihood is equal to

$$\mathbf{h}^T \mathbf{u}^0 + \sum_{i=1}^{N} \log f(y_i, \theta_0).$$

The second term is independent of $\mathbf{h}$, and therefore maximizing this function is equivalent to maximizing

$$\phi(\mathbf{u}^0, \mathbf{h}) \overset{\text{def}}{=} \mathbf{h}^T \mathbf{u}^0, \tag{4.7}$$

which is the statistic of Proposition 4.2 with $\alpha = 0$. Thus, the SIDE can be employed for finding the maximum likelihood hypothesis $\mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$, where $\mathbf{u}^0$ is related to the observation $\mathbf{y}$ through (4.6).

Example 4.4. *Changes in mean in a Gaussian random vector.*

In this example, $f(y,\theta_j)$ is the Gaussian density with mean θ_j and variance 1. We took $\theta_0 = 0$ and $\theta_1 = 1$. We assumed that the right number of edges, 10, is known, and so the stopping rule for the SIDE was $\nu_\alpha(t) \leq 10$. (In Subsection 4.4.3, we will treat the situation when the number of edges is a random variable, rather than a known parameter.)

The pointwise log-likelihoods (4.6) in this case are

$$u_i^0 = y_i - \frac{1}{2}(\theta_1 + \theta_0). \tag{4.8}$$

(Note that, if $\mathbf{u}(t)$ is the solution to the SIDE with the initial condition $\mathbf{u}^0$ of (4.8), and $\mathbf{u}'(t)$ is the solution to the SIDE with the initial condition $\mathbf{u}'(0) = \mathbf{y}$, then $\mathbf{u}'(t) = \mathbf{u}(t) + \alpha'$, where

$$\alpha' = \frac{1}{2}(\theta_1 + \theta_0), \tag{4.9}$$

and therefore the zero-crossings of $\mathbf{u}(t)$ coincide with the α'-crossings of $\mathbf{u}'(t)$. Consequently, we can simply evolve the SIDE with the data $\mathbf{y}$ as the initial condition, and look at its α'-crossings.)

Figure 4.3, from top down, depicts the true segmentation with ten edges, a corresponding observation $\mathbf{y}$, and the edges detected by the SIDE (the bottom plot will be explained in the next subsection). Note that the result is extremely accurate, despite the fact that the data is very noisy. The computations took 0.25 seconds on a Sparc Ultra 1, thanks to the fast implementation described in Subsection 4.3.1. ■

Example 4.5. *Changes in variance in a Gaussian random vector.*

Now $f(y,\theta_j)$ is a zero-mean Gaussian density with standard deviation θ_j; $\theta_0 = 1$ and $\theta_1 = 1.5$. The changes between the two are at the same locations as the jumps in the previous example (see the top plot of Figure 4.3). The top plot of Figure 4.4 shows an observation $\mathbf{y}$. Again, we assume that the number of changes is known. The bottom plot of Figure 4.4 shows the changes detected by the SIDE, depicted as a binary sequence of θ_0's and θ_1's. In addition to being very accurate, the computations took just 0.25 seconds. ■

■ 4.4.2 Two Gaussian Distributions with Unknown Means.

Suppose that $f(y,\theta_j)$ is the Gaussian density with mean θ_j and variance σ^2. Let $\mathbf{h}$ be an hypothesis, and let $\mathbf{Y}$ be a sequence of N random variables which are conditionally independent given $\mathbf{h}$, with the i-th random variable Y_i having conditional pdf $f(y, \theta_{h_i})$. Let ν be an upper bound on the number of edges in $\mathbf{h}$. Let K be the number of zeros in $\mathbf{h}$, and define $\sigma_1 = \frac{\sigma}{\theta_1 - \theta_0}\sqrt{N}$. Let the prior knowledge be as follows:

θ_0 and $\mathbf{h}$ are unknown;

σ, σ_1, and ν are known;

K is a random variable with the following discrete Gaussian probability mass function:

$$\mathrm{pr}(K = k) = C \exp\left\{ -\frac{1}{2}\left(\frac{k - \frac{N}{2}}{\sigma_1}\right)^2 \right\},\ k = 1, \ldots, N-1, \tag{4.10}$$

where C is a normalization constant.

We stress here that, even though our model assumes the knowledge of σ and σ_1, we will shortly see that the SIDE never uses them in computing the optimal hypothesis; hence the title of this subsection is justified. The only parameter required by the SIDE is ν. We also point out that the distribution (4.10) is a reasonable one: it assigns larger probabilities to the hypotheses with roughly as many zeros as ones. The reason for choosing this particular form of the distribution is to make the generalized likelihood simplify to the statistic of Proposition 4.2. This implies that the SIDE can be used to find the optimal solution. More precisely, given an observation $\mathbf{y}$ of $\mathbf{Y}$, we seek the best hypothesis in the generalized likelihood ratio sense [66]: the maximum likelihood estimates of the hypothesis and θ_0 are calculated for each value of K, and these estimates are then used in a multiple hypothesis testing procedure to estimate K. In other words, we seek

$$(\hat{\mathbf{h}}, \hat{\theta}_0, \hat{K}) = \arg\max_{\mathbf{h}, \theta_0, k}(\log f_1(\mathbf{y}|\mathbf{h}, \theta_0, k) + \log \mathrm{pr}(K = k)),$$

where f_1 is the conditional pdf of $\mathbf{Y}$. After simplifying this formula, we obtain that $\hat{\mathbf{h}}$ must maximize

$$\begin{aligned}\phi(\mathbf{y}, \mathbf{h}) &\stackrel{\text{def}}{=} \mathbf{h}^T\mathbf{y} - \frac{N-k}{N}\sum_{i=1}^{N} y_i \\ &= \mathbf{h}^T\left(\mathbf{y} - \mathbf{a}\left(\frac{1}{N}\sum_{i=1}^{N} y_i\right)\right),\end{aligned}$$

where $\alpha(x) = x$, and, as in Proposition 4.2, $\mathbf{a}(x) = (\alpha(x), \ldots, \alpha(x))^T \in I\!R^N$. Thus, according to Proposition 4.2, in order to find $\hat{\mathbf{h}}$, one has to evolve the SIDE whose initial condition is the observed signal: $\mathbf{u}^0 = \mathbf{y}$. The α-crossings of the solution $\mathbf{u}(t)$ will then coincide with the optimal edges, where

$$\alpha = \frac{1}{N}\sum_{i=1}^{N} u_i(t). \tag{4.11}$$

Thus, the only difference from Example 4.4 is that the threshold α' (4.9) is unknown, since $\frac{1}{2}(\theta_1 + \theta_0)$ is unknown. The threshold α (4.11) can be considered as an estimate of α'. If the underlying signal has roughly as many samples with mean θ_0 as ones with mean θ_1, then α is a good estimate of α', and we expect the estimates of the edge locations to be comparable to those in Example 4.4—i.e., despite less knowledge, the optimal estimates of the edge locations in this example would be similar to the optimal estimates of Example 4.4. This is confirmed by the experimental result for the data of Example 4.4, shown in the bottom plot of Figure 4.3, which is still very good and differs from the result of Example 4.4 in only two pixels out of the thousand. If the number of samples with mean θ_0 greatly differs from $\frac{N}{2}$, we would expect α to be a poor estimate of α', which will lead to larger errors in the optimal estimates of edge locations. This situation, however, is of low probability, according to our model (4.10).

■ 4.4.3 Random Number of Edges and the Mumford-Shah Functional.

As in the two previous subsections, let $\mathbf{y}$ be an observation of a sequence of N random variables, with conditional densities $p(y_i|\mathbf{h}) = f(y_i, \theta_{h_i})$, where $h_i \in \{0,1\}^N$. Given $\mathbf{h}$, the y_i's are conditionally independent. Let the number ν of edges in $\mathbf{h}$ be a random variable with a probability mass function $p_\nu(\bar{\nu})$, $\bar{\nu} = 0, 1, \ldots, N-1$. To find the maximum *a posteriori* estimate of ν, we need to maximize the conditional probability of ν, which depends on the unknown $\mathbf{h}$. We therefore, as in the previous subsection, use the generalized likelihood strategy [66]: the maximum likelihood estimates of $\mathbf{h}$ are calculated for each value of ν, and then these estimates are used in a multiple hypothesis test to estimate ν. In other words, we pick $\mathbf{h}$ to maximize

$$\sum_{i=1}^{N} \log p(y_i|\mathbf{h}) + \log p_\nu(\bar{\nu}). \tag{4.12}$$

We saw in Subsection 4.4.1 that the $\mathbf{h}$-dependent part of the likelihood term is $\mathbf{h}^T\mathbf{u}^0$, where $\mathbf{u}^0$ is the sequence of log-likelihood ratios. Therefore, maximizing (4.12) is equivalent to minimizing the following statistic $\psi(\mathbf{u}^0, \mathbf{h})$:

$$\psi(\mathbf{u}^0, \mathbf{h}) = -\mathbf{h}^T\mathbf{u}^0 - \log p_\nu(\bar{\nu}). \tag{4.13}$$

We now relate (4.13) to the SIDE.

Proposition 4.5. *Suppose that p_ν is monotonically non-increasing: $p_\nu(0) \geq p_\nu(1) \geq \ldots \geq p_\nu(N-1)$. Then the minimizer of $\psi(\mathbf{u}^0, \cdot)$ (4.13) can be computed using the SIDE, with post-processing whose computational complexity is $O(N)$.*

Proof. Let $\mathbf{h}_\psi$ be the hypothesis which minimizes $\psi(\mathbf{u}^0, \cdot)$, and, as previously, let

$$\mathbf{h}^*_{\leq\bar{\nu}}(\mathbf{u}^0), \text{ for } \bar{\nu} = 0, 1, \ldots, N-1 \tag{4.14}$$

be the hypothesis which achieves the maximal $\mathbf{h}^T\mathbf{u}^0$ among all the hypotheses with $\bar{\nu}$ or fewer edges. Suppose we could show that $\mathbf{h}_\psi$ is actually one of the hypotheses (4.14).

Then we could compute $\mathbf{h}_\psi$ as follows: run the SIDE to compute the N hypotheses (4.14), compute $\psi(\mathbf{u}^0, \cdot)$ for each of them, and pick the hypothesis which results in the smallest $\psi(\mathbf{u}^0, \cdot)$. To complete the proof, we now show that $\mathbf{h}_\psi$ is indeed one of the hypotheses (4.14).

Let us fix an arbitrary hypothesis $\bar{\mathbf{h}}$ with $\bar{\nu}$ edges, and let $\nu^* \leq \bar{\nu}$ be the number of edges in the hypothesis $\mathbf{h}^*_{\leq\bar{\nu}}(\mathbf{u}^0)$. Then, by the definition of $\mathbf{h}^*_{\leq\bar{\nu}}(\mathbf{u}^0)$, we have:

$$\left\{\mathbf{h}^*_{\leq\bar{\nu}}(\mathbf{u}^0)\right\}^T \mathbf{u}^0 \geq \bar{\mathbf{h}}^T \mathbf{u}^0. \tag{4.15}$$

The monotonicity of p_ν implies

$$\log p_\nu(\nu^*) \geq \log p_\nu(\bar{\nu}). \tag{4.16}$$

Summing the two inequalities (4.15) and (4.16), we get:

$$\psi(\mathbf{u}^0, \mathbf{h}^*_{\leq\bar{\nu}}(\mathbf{u}^0)) \leq \psi(\mathbf{u}^0, \bar{\mathbf{h}}).$$

In other words, for an arbitrary hypothesis $\bar{\mathbf{h}}$, we found an hypothesis from among (4.14) which results in a smaller (or equal) $\psi(\mathbf{u}^0, \cdot)$. Therefore, the optimal hypothesis $\mathbf{h}_\psi$ is among (4.14). ■

The second main result of this subsection is that having the exponential distribution p_ν is equivalent to specifying a stopping rule for the SIDE.

Proposition 4.6. *Let*

$$p_\nu(\bar{\nu}) = \frac{e^{-\lambda} - 1}{e^{-\lambda N} - 1} e^{-\lambda\bar{\nu}}, \text{ for } \bar{\nu} = 0, 1, \ldots, N-1, \tag{4.17}$$

and let $\mathbf{h}_\psi$ *be the hypothesis which minimizes* $\psi(\mathbf{u}^0, \cdot)$ *(4.13). Then the algorithm of Subsection 4.3.1, with a modified stopping rule, will produce* $\mathbf{h}_\psi$*. The new stopping criterion is:*

$$\min_{(i,j) \text{ is a region of } \mathbf{u}(t)} E_{ij}(t) > \lambda,$$

where $E_{ij(t)}$ *is defined as in (4.5).*

Proof. Substituting p_ν (4.17) into (4.13), we see that minimizing $\psi(\mathbf{u}^0, \mathbf{h})$ with respect to $\mathbf{h}$ is equivalent to minimizing the following function $\eta(\mathbf{h})$:

$$\eta(\mathbf{h}) = -\mathbf{h}^T \mathbf{u}^0 + \lambda\bar{\nu}. \tag{4.18}$$

Suppose that the solution to the SIDE has $\bar{\nu} + 1$ zero crossings at some time instant t, and call the hypothesis generated by these zero crossings $\mathbf{h}_1$. Let the next region to be removed be (i^*, j^*), and call the hypothesis resulting from its removal $\mathbf{h}_2$. Let us denote by $E^*(t)$ the energy of the region (i^*, j^*):

$$E^*(t) = \min_{(i,j) \text{ is a region of } \mathbf{u}(t)} E_{ij}(t).$$

In order to determine which hypothesis is better with respect to $\eta(\mathbf{h})$, we will look at $\eta(\mathbf{h}_2) - \eta(\mathbf{h}_1)$. First note that

$$(\mathbf{h}_2 - \mathbf{h}_1)^T \mathbf{u}^0 = - \left| \sum_{n=i^*}^{j^*} u_n^0 \right| = -\rho_{i^*,j^*} E^*(t),$$

and therefore

$$\eta(\mathbf{h}_2) - \eta(\mathbf{h}_1) = -(\mathbf{h}_2 - \mathbf{h}_1)^T \mathbf{u}^0 + \lambda(\bar{\nu} + 1 - \rho_{i^*,j^*}) - \lambda(\bar{\nu} + 1) = \rho_{i^*,j^*}(E^*(t) - \lambda).$$

So, if

$$E^*(t) > \lambda, \tag{4.19}$$

we prefer $\mathbf{h}_1$ to $\mathbf{h}_2$—i.e., it is better not to remove the region (i^*, j^*); otherwise, $\mathbf{h}_2$ is better than $\mathbf{h}_1$—i.e., we prefer to merge. Note that, by definition, $E^*(t)$ is a non-decreasing function of time, and so if (4.19) holds for some t, it will also hold for all future times. Also, if (4.19) is violated at some t, it is also violated for all past times. So, (4.19) is indeed a stopping rule.

The proof is not over, however, due to the fact that, as we remarked in Section 4.3, not every optimal hypothesis is generated by the SIDE. Therefore, we still need to rule out the possibility that the SIDE skips over the optimal hypothesis $\mathbf{h}_\psi$. In other words, we need to make sure that the following situation does not occur:

- $\mathbf{h}_\psi = \mathbf{h}^*_{\leq \bar{\nu}}(\mathbf{u}^0) \neq \mathbf{h}^*_{\leq \bar{\nu}-1}(\mathbf{u}^0)$, and
- $\rho_{i^*,j^*} = 2$, i.e., the SIDE's solution goes from $\bar{\nu}+1$ zero-crossings directly to $\bar{\nu}-1$.

The proof that this never happens is conceptually similar to the argument we made above when comparing $\mathbf{h}_1$ and $\mathbf{h}_2$. Since it is rather technical, we give it in Appendix B. The final result is that, if the solution to the SIDE goes from $\bar{\nu}+1$ zero-crossings directly to $\bar{\nu}-1$, skipping $\mathbf{h}^*_{\leq \bar{\nu}}(\mathbf{u}^0)$, then $\mathbf{h}^*_{\leq \bar{\nu}}(\mathbf{u}^0)$ cannot be $\mathbf{h}_\psi$. Therefore, the algorithm of Subsection 4.3.1 with the stopping rule (4.19) will find $\mathbf{h}_\psi$. ■

The importance of Proposition 4.6 stems from the fact that it provides the stopping rule for the SIDE (4.1) when the probabilistic model for the number of changes is (4.17). A more difficult question, which is not addressed in this thesis, is the following. Suppose it is known that the probability mass function of ν is of the form (4.17), but the parameter λ is free to be chosen. How can one adaptively select λ? One could approach this question by looking at the relative sizes of the energies $E^*(t)$ to decide at what time t the evolution starts to eliminate significant regions, rather than just those due to noise. This problem is conceptually similar to choosing the order of a model, and therefore various known techniques can be employed, like the minimum description length principle [53].

We conclude this section by showing that a specific conditional density $p(y_i|\mathbf{h})$ turns (4.18) into a special 1-D version of the Mumford-Shah functional [44]. It was proposed in [44] to estimate a piecewise-smooth function u from its noisy observations y by minimizing over u and Γ the following cost:

$$\frac{1}{2}\int_{I\!R^2}(u-y)^2+\gamma\int_{I\!R^2\setminus\Gamma}|\nabla u|^2+\lambda\bar{\nu}, \tag{4.20}$$

where Γ are the edges, i.e. the set on which u is discontinuous; $\bar{\nu}$ is the total length of the edges; and γ and λ are constants which control the smoothness of u within regions and the total length of the edges, respectively. If an approximation u is sought which is constant within each region [36,43], the second term disappears. In 1-D, the integration is over $I\!R^1$, and $\bar{\nu}$ is simply the number of the discontinuities in u. Assuming that we seek a piecewise-constant approximation, we discretize the 1-D version of (4.20):

$$\frac{1}{2}(\mathbf{u}-\mathbf{y})^T(\mathbf{u}-\mathbf{y})+\lambda\bar{\nu}. \tag{4.21}$$

If one is looking for a binary approximation $\mathbf{u}=\mathbf{h}\in\{0,1\}^N$, then $\mathbf{h}^T\mathbf{h}=\sum_{i=1}^N h_i$, and so if we define

$$u_i^0=y_i-\frac{1}{2}, \tag{4.22}$$

then minimizing (4.21) is equivalent to minimizing $\eta(\mathbf{h})$ (4.18). We note that (4.22) defines the log-likelihood ratios for the situation when $p(y_i|\mathbf{h})$ is the Gaussian density with unit variance and mean h_i. Indeed, in this case

$$\begin{aligned}\log p(y_i|h_i=1)-\log p(y_i|h_i=0) &= -\frac{1}{2}(y_i-1)^2+\frac{1}{2}y_i^2\\ &= y_i-\frac{1}{2}.\end{aligned}$$

We have just shown the following.

Proposition 4.7. *If*

$$\begin{aligned}p_\nu(\bar{\nu}) &= \frac{e^{-\lambda}-1}{e^{-\lambda N}-1}e^{-\lambda\bar{\nu}},\ \textit{for}\ \bar{\nu}=0,1,\ldots,N-1,\ \textit{and}\\ p(y_i|\mathbf{h}) &= \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(y_i-h_i)^2},\ \textit{for}\ i=1,\ldots,N,\end{aligned}$$

then the generalized likelihood function is (4.18), which is
a) a special case of the Mumford-Shah functional for 1-D signals, and
b) according to Proposition 4.6, is optimized by the SIDE. ■

4.5 Alternative Implementations.

4.5.1 Dynamic Programming.

The problem which is solved by the SIDE—namely, that of finding the optimal hypothesis $\mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$ with respect to the statistic ϕ of Proposition 4.2—can also be solved using dynamic programming. The basic idea is to scan the samples of $\mathbf{u}^0$ from left to right, and notice that the best hypothesis for the first $i+1$ samples can be easily calculated from the best hypothesis for the first i samples. More precisely, let $\mathbf{u}^{0i} = (u_1^0, u_2^0, \ldots, u_i^0)^T$ be the first i samples of $\mathbf{u}^0$. Let $\phi(\mathbf{u}^{0i}, \mathbf{h}) = \mathbf{h}^T(\mathbf{u}^{0i} - \mathbf{a})$, where $\mathbf{h} \in \{0,1\}^i$ and $\mathbf{a} = (\alpha, \ldots, \alpha)^T \in \mathbb{R}^i$. Let $A_{i\bar{\nu}p}$, for $i = 1, \ldots, N$, $\bar{\nu} = 0, \ldots, i-1$, and $p = 0, 1$, be the set of all hypotheses $\mathbf{h} \in \{0,1\}^i$ which have exactly $\bar{\nu}$ edges, and which end with p (i.e., $h_i = p$). Let $\mathbf{h}_{i\bar{\nu}p}$ be the best hypothesis in the set $A_{i\bar{\nu}p}$. Since ϕ is additive, $\mathbf{h}_{i+1,\bar{\nu},p}$ is either $(\mathbf{h}^T_{i\bar{\nu}p}, p)^T$ or $(\mathbf{h}^T_{i,\bar{\nu}-1,1-p}, p)^T$. Therefore, once we have computed and stored $\mathbf{h}_{i\bar{\nu}p}$ and the corresponding values of ϕ for $\bar{\nu} = 0, \ldots, \nu$, and $p = 0, 1$, it will take us one comparison to compute $\mathbf{h}_{i+1,\bar{\nu},p}$ for each $\bar{\nu}$ and p. Thus, to go from $\mathbf{h}_{i,\bar{\nu},p}$ to $\mathbf{h}_{i+1,\bar{\nu},p}$, we will need $O(\nu)$ computations. The total computational complexity is therefore $O(\nu N)$. The required memory is also $O(\nu N)$. Thus, if ν is smaller than $\log N$, the dynamic programming approach is faster than the SIDE implemented as the region merging method of Subsection 4.3.1. If ν is comparable to $\log N$, the two methods are similar in speed, but the SIDE wins in terms of memory use, since it only requires $O(N)$ memory. Finally, if ν is larger than $\log N$, the SIDE wins both in complexity and memory. Another important advantage of the SIDE is that it generalizes to 2-D—as shown in Section 4.7, while the dynamic programming algorithm does not.

4.5.2 An Equivalent Linear Program.

We now show that the problem of minimizing η (4.18), considered in Section 4.4, is a linear programming problem.

Let D be the $(N-1)$-by-N first-difference matrix. Then, for any hypothesis $\mathbf{h} \in \{0,1\}^N$, we have that the ℓ^1-norm of its first difference $\|D\mathbf{h}\|_1$ is exactly equal to the number of edges in $\mathbf{h}$. Therefore, we can re-write the problem of minimizing η (4.18) as follows:

$$\begin{aligned} \text{minimize} \quad & -\mathbf{h}^T\mathbf{u}^0 + \lambda\|D\mathbf{h}\|_1, \\ \text{subject to} \quad & \mathbf{h} \in \{0,1\}^N. \end{aligned} \tag{4.23}$$

This problem is equivalent to the following linear program:

$$\begin{array}{llll} \text{minimize} & -\mathbf{h}^T\mathbf{u}^0 + \lambda\sum_{i=1}^{N-1} r_i & & (4.24) \\ \text{subject to} & r_i \geq h_{i+1} - h_i & \text{for } i = 1, \ldots, N-1 & (4.25) \\ & r_i \geq h_i - h_{i+1} & \text{for } i = 1, \ldots, N-1 & (4.26) \\ & 0 \leq h_i \leq 1 & \text{for } i = 1, \ldots, N, & (4.27) \end{array}$$

where we replaced the requirement that $\mathbf{h}$ be binary with a seemingly less restrictive condition that $\mathbf{h}$ belong to the unit hypercube $[0,1]^N$. Constraints (4.25) and (4.26) mean that $r_i \geq |h_{i+1} - h_i|$. On the other hand, (4.24) means that r_i must be as small as possible, and therefore $r_i = |h_{i+1} - h_i|$. The fact that the constraints (4.27) are equivalent to $\mathbf{h} \in \{0,1\}^N$ is a little less obvious; it is verified in Appendix B (Section B.5). We point out that any generic linear programming algorithm will lose out in speed to the SIDE, because the SIDE exploits the special structure of the problem (4.23).

4.6 Performance Analysis.

This section is devoted to the performance analysis of the SIDE, for signals with one change in mean. We first consider the event that the detected change location is farther than a certain number of samples from the actual one. An expression for the lower bound of the probability of this event is derived in Subsection 4.6.1; it is applied to the analysis of signals with additive white Gaussian noise in Subsection 4.6.2. We address the robustness of our algorithm in Subsection 4.6.3, by showing its optimality with respect to a certain H_∞-like performance criterion.

4.6.1 Probability Bounds.

This section is devoted to the analysis of the example in Subsection 4.4.2, when the number of edges is equal to one. That is, in order to maximize

$$\phi(\mathbf{u}^0, \mathbf{h}) = \mathbf{h}^T\mathbf{u}^0 - \frac{N-k}{N}\sum_{i=1}^{N} u_i^0,$$

we evolve the SIDE (4.1) until exactly one α-crossing remains, where $\alpha = \frac{1}{N}\sum_{i=1}^N u_i^0$. We denote the correct hypothesis $\mathbf{h}^c$ and the correct location of the edge g^c. Without loss of generality, we assume that the first g^c samples of $\mathbf{u}^0$ have mean θ_0, the last $N - g^c$ samples have mean θ_1, and that $d \overset{\text{def}}{=} \theta_1 - \theta_0 > 0$. We denote the detected edge location g^*, its sign z^*, the corresponding hypothesis $\mathbf{h}^*$, and the number of zeros in it k^*: if $z^* = 1$, then $k^* = g^*$; if $z^* = -1$, then $k^* = N - g^*$.

Pick two integers, p_0 and q_0, satisfying $1 \leq p_0 \leq g^c \leq q_0 \leq N-1$, so that the location g^c of the true edge is between p_0 and q_0. The goal of this section is to compute a lower bound for the probability of the event

$$\{p_0 \leq g^* \leq q_0, z^* = 1\}, \tag{4.28}$$

which says that the detected edge location g^* is between p_0 and q_0, and that the detected sign of the edge is correct. The strategy will be to find a lower bound for the probability of a simpler event which implies (4.28). Specifically, suppose that $g^* > g^c$ and $z^* = 1$. Then

$$\phi(\mathbf{u}^0, \mathbf{h}^*) - \phi(\mathbf{u}^0, \mathbf{h}^c) = \sum_{i=g^c+1}^{g^*} (-u_i^0 + \alpha).$$

Since $\mathbf{h}^*$ is the optimal hypothesis, the above expression has to be positive. Thus, if

$$\sum_{i=g^c+1}^{q} (-u_i^0 + \alpha) < 0 \text{ for } q = q_0 + 1, \dots, N, \tag{4.29}$$

then g^* cannot be larger than q_0. Similarly, if

$$\sum_{i=p+1}^{g^c} (u_i^0 - \alpha) < 0 \text{ for } p = 1, \dots, p_0 - 1, \tag{4.30}$$

then g^* cannot be smaller than p_0. Similar arguments show that, if

$$\sum_{i=g^c+1}^{N} (-u_i^0 + \alpha) + \sum_{i=q}^{N} (-u_i^0 + \alpha) < 0 \quad \text{for} \quad q = g^c + 1, \dots, N, \text{ and} \tag{4.31}$$

$$\sum_{i=1}^{g^c} (u_i^0 - \alpha) + \sum_{i=1}^{p} (u_i^0 - \alpha) < 0 \quad \text{for} \quad p = 1, \dots, g^c, \tag{4.32}$$

then the detected sign is correct, i.e. $z^* = 1$. Thus, the simultaneous occurrence of the events (4.29)-(4.32) implies (4.28). If α were not random, the expressions in (4.29)-(4.32) would be sums of independent identically distributed random variables, and therefore we would be able to employ results from the theory of random walks. We will remove the randomness of α from (4.29)-(4.32) by introducing a non-random bound on how far α can be from its mean

$$m \stackrel{\text{def}}{=} \frac{1}{N}(g^c\theta_0 + (N - g^c)\theta_1).$$

In other words, suppose that there are two positive real numbers, δ_1 and δ_2, such that

$$m - \delta_1 \leq \alpha \leq m + \delta_2. \tag{4.33}$$

Then each of the inequalities

$$\sum_{i=g^c+1}^{q} (-u_i^0 + m + \delta_2) < 0 \text{ for } q = q_0 + 1, \dots, N \tag{4.34}$$

implies the corresponding inequality in (4.29). Let us call A_q the event that the q-th inequality in (4.34) holds, for $q = q_0 + 1, \dots, N$. We shall similarly bound the events (4.30), by defining events A_p whose intersection implies (4.30):

$$A_p = \left\{ \sum_{i=p+1}^{g^c} (u_i^0 - (m - \delta_1)) < 0 \right\}, \text{ for } p = 1, \dots, p_0 - 1. \tag{4.35}$$

We shall call A'_q and A'_p the events which imply (4.31) and (4.32), respectively:

$$\begin{aligned} A'_q &= \left\{ \left(\sum_{i=g^c+1}^{N} + \sum_{i=q}^{N} \right) (-u_i^0 + m + \delta_2) < 0, \right\} \text{ for } q = g^c + 1, \dots , N; && (4.36) \\ A'_p &= \left\{ \left(\sum_{i=1}^{g^c} + \sum_{i=1}^{p} \right) (u_i^0 - (m - \delta_1)) < 0 \right\}, \text{ for } p = 1, \dots , g^c. && (4.37) \end{aligned}$$

Let ε_1 be the union (upper) bound for the probability of $\cup_{p=1}^{g^c} \overline{A'_p}$, where the overbar denotes the complement:

$$\varepsilon_1 = \sum_{p=1}^{g^c} \Pr(\overline{A'_p}). \tag{4.38}$$

Suppose further that p_1 is a lower bound for the probability of the intersection of the events A_p:

$$p_1 \leq \Pr \left(\bigcap_{p=1}^{p_0 - 1} A_p \right). \tag{4.39}$$

Then

$$\begin{aligned} \Pr \left(\bigcap_{i=1}^{p_0-1} A_p \cap \left(\bigcap_{i=1}^{g^c} A'_p \right) \right) &= \Pr \left(\bigcap_{i=1}^{p_0-1} A_p \cap \left(\overline{\bigcup_{i=1}^{g_c} \overline{A'_p}} \right) \right) && (4.40) \\ &= \Pr \left(\bigcap_{i=1}^{p_0-1} A_p \right) - \Pr \left(\bigcap_{i=1}^{p_0-1} A_p \cap \left(\bigcup_{i=1}^{g_c} \overline{A'_p} \right) \right) && (4.41) \\ &\geq p_1 - \Pr \left(\bigcup_{i=1}^{g_c} \overline{A'_p} \right) && (4.42) \\ &\geq p_1 - \varepsilon_1, && (4.43) \end{aligned}$$

where we used the identity $\overline{\cap A'_p} = \cup \overline{A'_p}$ in (4.40), the identity $\Pr(A \cap B) = \Pr(A) - \Pr(A \cap \overline{B})$ in (4.41), and the inequality $-\Pr(A \cap \overline{B}) \geq -\Pr(\overline{B})$ in (4.42). Similarly, the probability of the intersection of (4.34) and (4.36) is bounded from below by $p_2 - \varepsilon_2$, where

$$\varepsilon_2 = \sum_{q=g^c+1}^{N} \Pr(\overline{A'_q}),$$

$$p_2 \leq \Pr \left(\bigcap_{q=q_0+1}^{N} A_q \right).$$

Finally, if we denote by $1-\varepsilon$ the probability of (4.33), we have that the intersection of the events (4.33)-(4.37) is lower-bounded by

$$(p_1-\varepsilon_1)(p_2-\varepsilon_2)-\varepsilon. \tag{4.44}$$

We showed earlier in this section that the intersection of these events implies the intersection of the events (4.29)-(4.32), which, in turn, implies the event (4.28). Thus, the above expression (4.44) is a lower bound for the probability of the event (4.28).

In [28], asymptotic probabilities of the events (4.28) are computed, for $N \to \infty$ and $g^c \to \infty$. When α is non-random (as in, e.g., our Examples 4.4 and 4.5 of Subsection 4.4.1), these asymptotic probabilities are also (non-asymptotic) lower bounds. In the process of computing these, lower bounds p_1 (4.39) and p_2 are also computed in [28]; these are asymptotically tight. In the next subsection, we describe a different method for computing p_1 and p_2 for the Gaussian case (i.e. when the model of Subsection 4.4.2 applies). Our method produces looser bounds than [28]; however, the derivation is conceptually much simpler and leads to easier computations.

■ 4.6.2 White Gaussian Noise.

Proposition 4.8. *Suppose that the following model applies:*

$$\mathbf{u}^0 = \mathbf{m} + \mathbf{w},$$

where $\mathbf{m} = (\theta_0, \dots, \theta_0, \theta_1 \dots, \theta_1)^T$ *is the mean vector of* $\mathbf{u}^0$*, and* $\mathbf{w}$ *is zero-mean white Gaussian noise with variance* σ^2*. Then*

$$\begin{aligned}
&Pr(p_0 \le g^* \le q_0 \text{ and } z^* = 1) \ge \\
&\left\{\Phi\left(\frac{d_1\sqrt{g^c-p_0+1}}{\sigma}\right) - \Phi\left(-\frac{d_1\sqrt{g^c-p_0+1}}{\sigma}\right) - \sum_{p=1}^{g^c}\Phi\left(-\frac{d_1(p+g^c)}{\sigma\sqrt{3p+g^c}}\right)\right\} \times \\
\times &\left\{\Phi\left(\frac{d_2\sqrt{q_0-g^c+1}}{\sigma}\right) - \Phi\left(-\frac{d_2\sqrt{q_0-g^c+1}}{\sigma}\right) - \sum_{q=g^c+1}^{N}\Phi\left(-\frac{d_2(2N-q-g^c)}{\sigma\sqrt{4N-3q-g^c}}\right)\right\} - \\
- \; &\Phi\left(-\frac{\delta_2\sqrt{N}}{\sigma}\right) - \Phi\left(-\frac{\delta_1\sqrt{N}}{\sigma}\right),
\end{aligned} \tag{4.45}$$

where

- Φ *is the Gaussian cumulative distribution function;*
- $d_1 = \frac{(N-g^c)d}{N} - \delta_1$*;*
- $d_2 = \frac{g^c d}{N} - \delta_2$*;*
- δ_1 *and* δ_2 *are any positive real numbers such that* $d_1 > 0$ *and* $d_2 > 0$*.*

Proof. The terms of this bound come from calculating the parameters p_1, p_2, ε_1, ε_2, and ε of the expression (4.44). We now present this calculation.

It follows from our noise model that $\alpha = \frac{1}{N}\sum_{i=1}^{N} u_i^0$ is Gaussian with mean $\frac{1}{N}(g^c\theta_0 + (N-g^c)\theta_1)$ and variance $\frac{\sigma^2}{N}$. Thus, the probability ε that (4.33) does not hold is

$$\varepsilon = 1 - \int_{-\delta_1}^{\delta_2} \mathcal{N}\left(0, \frac{\sigma^2}{N}\right) = \Phi\left(-\frac{\delta_1\sqrt{N}}{\sigma}\right) + \Phi\left(-\frac{\delta_2\sqrt{N}}{\sigma}\right), \tag{4.46}$$

where $\mathcal{N}$ is the Gaussian density.

In order to find ε_1, re-write the event $\overline{A'_p}$ as

$$2\sum_{i=1}^{p} w_i + \sum_{i=p+1}^{g^c} w_i - (p+g^c)\left(\frac{(N-g^c)d}{N} - \delta_1\right) > 0.$$

The sum of the noise samples is a zero-mean Gaussian random variable with variance $\sigma^2(4p + (g^c - p)) = \sigma^2(3p + g^c)$. Therefore, if we define

$$d_1 = \frac{(N-g^c)d}{N} - \delta_1,$$

then the probability of $\overline{A'_p}$ is

$$\Phi\left(-\frac{d_1(p+g^c)}{\sigma\sqrt{3p+g^c}}\right).$$

Substituting this into (4.38), we get:

$$\varepsilon_1 = \sum_{p=1}^{g^c} \Phi\left(-\frac{d_1(p+g^c)}{\sigma\sqrt{3p+g^c}}\right).$$

To find a lower bound p_1 (see Equation (4.39)), we define $S_0 = 0$ and

$$S_j = \frac{1}{\sigma}\sum_{i=g^c-j+1}^{g^c} w_i \text{ for } j = 1, \dots, g^c,$$

and note that the intersection of the events A_p of Equation (4.35) is equivalent to

$$S_j < \frac{d_1}{\sigma} j \text{ for } j = g^c - p_0 + 1, \dots, g^c - 1.$$

Also note that the S_j's form the standard (discrete) Brownian motion [52], which can be viewed as a sampling of the standard continuous Brownian motion $S(t)$ at integer times. We define $t_0 = g^c - p_0 + 1$, $s_0 = S(t_0)$, and $f(s_0) =$ the pdf of s_0. Then

$$\begin{aligned}
\Pr\left(\bigcap_{p=1}^{p_0-1} A_p\right) &= \Pr(S_j < \frac{d_1}{\sigma} j \text{ for } j = t_0, ..., g^c - 1) \\
&\geq \Pr(S(t) < \frac{d_1}{\sigma} t \text{ for } t_0 \leq t \leq g^c - 1) \\
&= \int_{-\infty}^{\frac{d_1}{\sigma} t_0} \Pr\left(S(t) < \frac{d_1}{\sigma} t \text{ for } t > t_0 | s_0\right) f(s_0) ds_0.
\end{aligned}$$

Conditioned on s_0, the process $P(t) = S(t + t_0) - s_0$ is the standard Brownian motion for $t \geq 0$. Therefore, the integral above is equal to

$$\int_{-\infty}^{\frac{d_1}{\sigma} t_0} \Pr\left(P(t) < \frac{d_1}{\sigma}(t + t_0) - s_0 \text{ for } t > 0 | s_0\right) f(s_0) ds_0 =$$

$$\int_{-\infty}^{\frac{d_1}{\sigma} t_0} \Pr\left(\sup_{t \geq 0}(P(t) - \frac{d_1}{\sigma} t) < \frac{d_1}{\sigma} t_0 - s_0 | s_0\right) f(s_0) ds_0.$$

Given s_0, $P(t) - \frac{d_1}{\sigma} t$ is a Brownian motion with drift $-\frac{d_1}{\sigma}$. If the drift is non-negative, then the probability inside the integral is zero [52]. We therefore assume that $d_1 > 0$, i.e., that

$$\delta_1 < \frac{(N - g^c)d}{N}.$$

Then the drift is negative, in which case the supremum is finite almost surely, and its probability distribution is [52]

$$1 - \exp\left(-2\frac{d_1}{\sigma} x\right) \text{ for } x \geq 0, \text{ and zero otherwise.}$$

Substituting this into the integral above, we get

$$\int_{-\infty}^{\frac{d_1}{\sigma} t_0} \left\{1 - \exp\left[-2\frac{d_1}{\sigma}\left(\frac{d_1}{\sigma} t_0 - s_0\right)\right]\right\} \frac{1}{\sqrt{2\pi t_0}} \exp\left(-\frac{s_0^2}{2t_0}\right) ds_0 =$$

$$\Phi\left(\frac{d_1\sqrt{t_0}}{\sigma}\right) - \int_{-\infty}^{\frac{d_1}{\sigma} t_0} \frac{1}{\sqrt{2\pi t_0}} \exp\left(-\frac{(s_0 - \frac{2d_1 t_0}{\sigma})^2}{2t_0}\right) ds_0 =$$

$$\Phi\left(\frac{d_1\sqrt{g^c - p_0 + 1}}{\sigma}\right) - \Phi\left(-\frac{d_1\sqrt{g^c - p_0 + 1}}{\sigma}\right).$$

Combining the values for p_1, ε_1, and ε, obtained above, with similarly obtained values for p_2 and ε_2 (where $d_2 = \frac{g^c d}{N} - \delta_2$), we arrive at the expression (4.45). As we mentioned above, this bound is looser than those of [28]. For example, if N is very large, $\frac{g^c}{N} = 0.5$, $d = 3\sigma$, and $p_0 = q_0 = g^c$, then the asymptotic probability (from Table 3.3 of [28]) is 0.857, whereas our bound is 0.751.

■ 4.6.3 H_∞-Like Optimality.

We continue the analysis of detecting one change in mean in a sequence of Gaussian random variables, but now we assume that both means are known, and therefore $\alpha = 0$, as in Example 4.4 of Subsection 4.4.1. We now show that the change location estimate produced by the SIDE—which in this case, according to Section 4.3, is the maximum likelihood estimate—is optimal according to an H_∞-like criterion.

H_∞ estimation and control arose out of situations when there is no complete *a priori* knowledge of the system dynamics and of the statistical properties of the exogenous inputs. Model uncertainties and the lack of statistical information encountered in many applications have led to research in minimax estimation, producing so-called H_∞ algorithms which are robust to parameter variations [25, 26, 45, 61]. In these approaches, the quantity to be minimized is the H_∞ norm of the operator mapping the inputs to the desired function (either the output or a weighted error).

We will analyze the following problem.

Problem 4.2. *Let $d > 0$ be a known real number. Consider a step sequence $\mathbf{m}_{g^c} = (0, \ldots, 0, d \ldots, d)^T$ of length N, whose first g^c entries are zeros. Let the observed signal be*

$$\mathbf{y} = \mathbf{m}_{g^c} + \mathbf{v},$$

where $\mathbf{v}$ is an unknown disturbance. The objective is to estimate the location of change $g^c \in \{0, \ldots, N\}$ ($g^c = 0$ and $g^c = N$ refer to the same hypothesis, corresponding to no change). ■

As stated in Section 4.3, if $\mathbf{v}$ is a zero-mean white Gaussian noise, then the SIDE will find the ML estimate, i.e.

$$\hat{g}^c_{ML} = \arg\max_g \sum_{i=g+1}^{N} (y_i - \frac{d}{2}). \tag{4.47}$$

To analyze the robustness of this estimator, we define the following performance measure:

$$\mathcal{B}(f) = \sup_{g^c, \mathbf{v} \neq \mathbf{0}} \frac{|g^c - f(\mathbf{y})|}{\|\mathbf{v}\|_1},$$

where $f(\mathbf{y})$ is any estimator of g^c, and $\|\cdot\|_1$ stands for the ℓ^1 norm. Choosing the estimator which minimizes $\mathcal{B}$ is similar in spirit to H_∞ estimation: we would like to minimize the worst possible error, over all possible disturbances. We presently show that the SIDE estimator (4.47) does minimize $\mathcal{B}$. This means that our estimator is robust: it has the best worst-case error performance among all estimators, and for all noise sequences.

We will prove the optimality of the SIDE estimator by showing two things:

- that $\mathcal{B}(f)$ is always larger than a certain constant (see Proposition 4.9 below), and
- that the SIDE estimator achieves this lower bound (see Proposition 4.10 below).

Proposition 4.9. *For any estimator* f, $\mathcal{B}(f) \geq \frac{2}{d}$.

Proof. Fix the noise level at $\|\mathbf{v}\|_1 = \frac{d}{2}$, and suppose that the observation is $\mathbf{y} = (0, \frac{d}{2}, d, \ldots, d)^T$. The signal $\mathbf{m}_{g^c}$ which resulted in this $\mathbf{y}$ after adding noise of norm $\frac{d}{2}$ could be either $\mathbf{m}_1 = (0, d, d, \ldots, d)^T$ or $\mathbf{m}_2 = (0, 0, d, \ldots, d)^T$, in which cases $g^c = 1$ and $g^c = 2$, respectively. Thus,

$$\begin{aligned}
\mathcal{B}(f) &\geq \sup_{g^c \in \{1,2\}} \frac{|g^c - f(\mathbf{y})|}{\frac{d}{2}} = \\
&= \frac{2}{d} \sup_{g^c \in \{1,2\}} |g^c - f(\mathbf{y})| = \\
&= \begin{cases} \frac{2}{d}|2 - f(\mathbf{y})| & \text{if} \quad f(\mathbf{y}) = 1 \\ \frac{2}{d}|1 - f(\mathbf{y})| & \text{if} \quad f(\mathbf{y}) \geq 2 \end{cases} \geq \\
&\geq \frac{2}{d}. \quad \blacksquare
\end{aligned}$$

We will now show that the ML estimator achieves this bound.

Proposition 4.10. *For* $\hat{g}^c_{ML} = f_{ML}(\mathbf{y})$, $\mathcal{B}(f_{ML}) = \frac{2}{d}$. *Thus, the estimator* $\hat{g}^c_{ML}$ *is optimal with respect to the criterion* $\mathcal{B}$.

Proof. Suppose that $\hat{g}^c_{ML} > g^c$. Then (4.47) implies

$$\begin{aligned}
\sum_{i=\hat{g}^c_{ML}+1}^{N} (y_i - \frac{d}{2}) &> \sum_{i=g^c+1}^{N} (y_i - \frac{d}{2}) \Rightarrow \\
\sum_{i=g^c+1}^{\hat{g}^c_{ML}} (y_i - \frac{d}{2}) &< 0 \Rightarrow \\
\sum_{i=g^c+1}^{\hat{g}^c_{ML}} ((d + v_i) - \frac{d}{2}) &< 0 \Rightarrow \\
(\hat{g}^c_{ML} - g^c)\frac{d}{2} &< -\sum_{i=g^c+1}^{\hat{g}^c_{ML}} v_i \leq \|\mathbf{v}\|_1.
\end{aligned}$$

Therefore, the smallest ℓ^1 norm of the disturbance required to create the error $\hat{g}^c_{ML} - g^c$ is $(\hat{g}^c_{ML} - g^c)\frac{d}{2}$. The calculations for $\hat{g}^c_{ML} < g^c$ are similar, and so

$$\mathcal{B}(f_{ML}) = \sup_{g^c, \mathbf{v} \neq 0} \frac{|g^c - f_{ML}(\mathbf{y})|}{\|\mathbf{v}\|_1} \leq \sup_{g^c} \frac{|g^c - f_{ML}(\mathbf{y})|}{|g^c - f_{ML}(\mathbf{y})|\frac{d}{2}} = \frac{2}{d}. \quad \blacksquare$$

■ 4.7 Analysis in 2-D.

In 1-D, there are always one fewer edges than regions. In 2-D, there is no such relationship between the total length of the boundaries and the total number of regions. From this difference stem two ways to generalize the SIDE (4.1) to 2-D. The first one is to evolve the intensity value inside each region according to

$$\dot{u}_i = \frac{1}{m_i}\operatorname{sgn}(\alpha - u_i)p_i, \tag{4.48}$$

where m_i is the number of pixels inside the i-th region, and p_i is the perimeter of the i-th region. (We still define the boundaries between regions as lines of α-crossings.) The second possible 2-D generalization of the SIDE has the same form, with p_i being the number of neighboring regions of the region i.

The 2-D SIDEs unfortunately do not find the global solution to the optimization problem of Proposition 4.2. However, we believe that a similar but weaker statement can be made—namely, that they find optimal coarsenings of the initial segmentation. We briefly repeat the definitions from Chapter 2 which are used in this statement.

A *segmentation* of a given image $\mathbf{u}^0$ is a partitioning of the domain Ω of its definition into several disjoint regions $O_1, \ \dots \ , O_k$ ($\cup_{i=1}^{k} O_i = \Omega$). If a segmentation $E_1 = \cup O'_j$ can be obtained from a segmentation $E = \cup O_i$ by erasing edges, it is said that E_1 is *coarser* than E. More precisely, E_1 is *coarser* than E when for every region O_i of E there is a region O'_j of E_1 such that $O_i \subset O'_j$.

Conjecture 1. *As in Proposition 4.2, let* $\phi(\mathbf{u}, \mathbf{h}) = \mathbf{h}^T(\mathbf{u} - \mathbf{a})$. *Let* E *be the segmentation of* $\mathbf{u}^0$ *given by the* α*-crossings of* $\mathbf{u}^0$. *Consider the SIDE (4.48) with* $\mathbf{u}^0$ *as the initial data and with* p_i *equal to the perimeter of the* i*-th region. Suppose the evolution is stopped when the total perimeter of the* α*-crossings is* ν, *and let* E_1 *be the segmentation given by these* α*-crossings. Then* E_1 *is the optimal coarsening of* E, *subject to the constraint that the segmentation have total edge length* ν *or less.* ■

Conjecture 2. *The situation is the same as in Conjecture 1, except that* p_i *is the number of neighbors of the* i*-th region. Suppose the evolution is stopped when the total number of regions is* ν, *and let* E_1 *be the resulting segmentation. Then* E_1 *is the optimal coarsening of* E, *subject to the constraint that the segmentation have* ν *or fewer regions.* ■

We note that, while (4.48) is the equation we have conjectures about, it is not the equation we use in practice for images. The reason is that thresholding the initial condition $\mathbf{u}^0$ with the threshold α typically leads to initial segmentations which are too coarse—that is, which have too few regions. Even if the evolution then provides the best coarsening of this initial segmentation, it may not be a good result. Better results are achieved when one first evolves the SIDE using (3.12), and then applies the threshold α to the image $\mathbf{u}(t)$. This is what was done in examples of Figure 4.5, which are experimental evidence of the fact that the algorithm works well and is very robust to degradations which do not conform well to the models of Section 4.3. The data on the left is a very blurry and noisy synthetic aperture radar image of two textures: forest

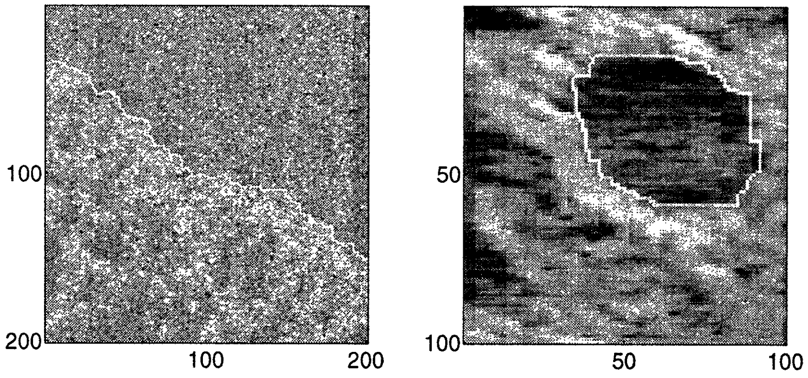


Figure 4.5. Edge detection in 2-D.

and grass. The pervasive speckle noise is inherent to this method of data collection. The algorithm was run on the raw data itself (which corresponds to assuming a Gaussian model with changes in mean—see Section 4.3), and stopped when two regions remained. The resulting boundary (shown superimposed onto the logarithm of the original image) is extremely accurate. The logarithm of a similarly blurry and noisy ultrasound image of a thyroid is shown on the right, with the boundary detected by the SIDE.

Chapter 5

Segmentation of Color, Texture, and Orientation Images

THE preceding chapters were all devoted to the analysis of images and signals which take values in $\mathbb{R}$. It is often necessary, however, to process vector-valued images where each pixel value is a vector belonging to $\mathbb{R}^M$, with $M \geq 1$. The entries of this vector could correspond to red, green, and blue intensity values in color images [74], to data gathered from several sensors [33] or imaging modalities [68], or to the entries of a feature vector obtained from analyzing a texture image [10]. In the next section, we generalize our SIDEs to vector-valued images and argue that most properties of scalar-valued SIDEs still apply. We then give several examples of segmentation of color and texture images.

Section 5.2 treats images whose every pixel belongs to a circle $\mathbb{S}^1$. Such images arise in the analysis of orientation [48] and optical flow [30].

■ 5.1 Vector-Valued Images.

We use one of the properties discussed in Chapter 3 in order to generalize our evolution equations to vector-valued images. We recall that a SIDE is the gradient descent for the global energy (2.8), which we re-write as follows:

$$\mathcal{E} = \sum_{i,j \text{are neighbors}} E(\|u_j - u_i\|),$$

where E is a SIDE energy function (Figure 2.6). The norm $\|\cdot\|$ here stands simply for the absolute value of its scalar argument. Now notice that we still can use the above equation if the image under consideration is vector-valued, by interpreting $\|\cdot\|$ as the ℓ^2 norm. To remind ourselves that the pixel values of vector-valued images are vectors, we will use arrows to denote them:

$$\mathcal{E} = \sum_{i,j \text{are neighbors}} E(\|\vec{u}_j - \vec{u}_i\|), \qquad (5.1)$$

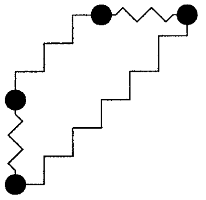


Figure 5.1. Spring-mass model for vector-valued diffusions. This figure shows a 2-by-2 image whose pixels are two-vectors: (2,2), (0,0), (0,1), and (1,2). The pixel values are depicted, with each pixel connected by springs to its neighboring pixels.

where

$$\begin{aligned}\vec{u}_i &= (u_{i,1}, \ldots, u_{i,M})^T \in I\!R^M, \\ \vec{u}_j &= (u_{j,1}, \ldots, u_{j,M})^T \in I\!R^M, \text{ and} \\ \|\vec{u}_j - \vec{u}_i\| &= \sqrt{\sum_{k=1}^{M} (u_{j,k} - u_{i,k})^2}.\end{aligned}$$

We will call the collection of the k-th entries of these vectors the *k-th channel* of the image. In this case, the gradient descent equation is:

$$\dot{\vec{u}}_i = \frac{1}{m_i} \sum_{j \in A_i} \frac{\vec{u}_j - \vec{u}_i}{\|\vec{u}_j - \vec{u}_i\|} F(\|\vec{u}_j - \vec{u}_i\|) p_{ij}. \tag{5.2}$$

(This notation combines M equations—one for each channel—in a single vector equation). Just as for the scalar images, we merge two neighboring pixels at locations i and j when their values become equal: $\vec{u}_j(t) = \vec{u}_i(t)$. Just as in Chapter 3, m_i and A_i are the area of the i-th region and the set of its neighbors, respectively. The length of the boundary between regions i and j is p_{ij}.

A slight modification of the spring-mass models of Chapter 3, Figures 3.1 and 3.3, can be used to visualize this evolution. We recall that those models consisted of particles forced to move along straight lines, and connected by springs to their neighbors. If we replace each straight line with an M-dimensional space, we will get the model corresponding to the vector-valued equation, with pixel values in $I\!R^M$. For example, when $M = 2$, the particles are forced to move in 2-D planes. Another way to visualize the system is by depicting all the particles as points in a single M-dimensional space, as shown in Figure 5.1. Each pixel value $\vec{u}_i = (u_{i,1}, u_{i,2})^T$ is depicted in Figure 5.1 as a particle whose coordinates are $u_{i,1}$ and $u_{i,2}$. Each particle is connected by springs to its neighbors. The spring whose length is v exerts a force whose absolute value is $F(v)$, and which is directed parallel to the spring.

By using techniques similar to those in Chapter 3, it can be verified that Equation (5.2) inherits many useful properties of the scalar equation. Namely, the conservation of mean and the local maximum principle hold for each channel; the equation reaches

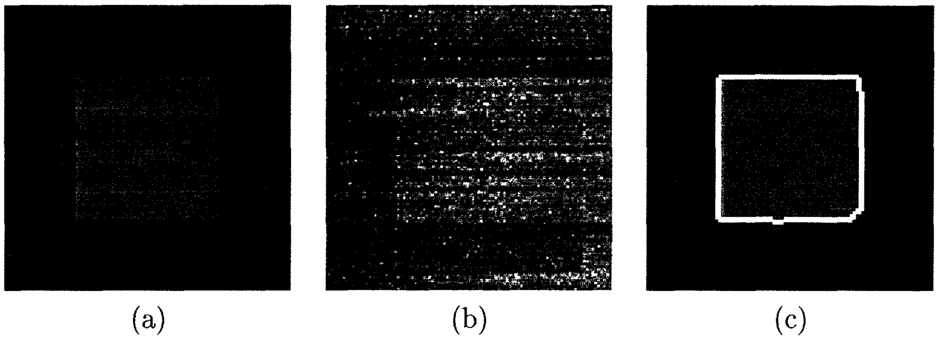


Figure 5.2. (a) A test image; (b) its noisy version (normalized); (c) detected boundary, superimposed onto the noise-free image

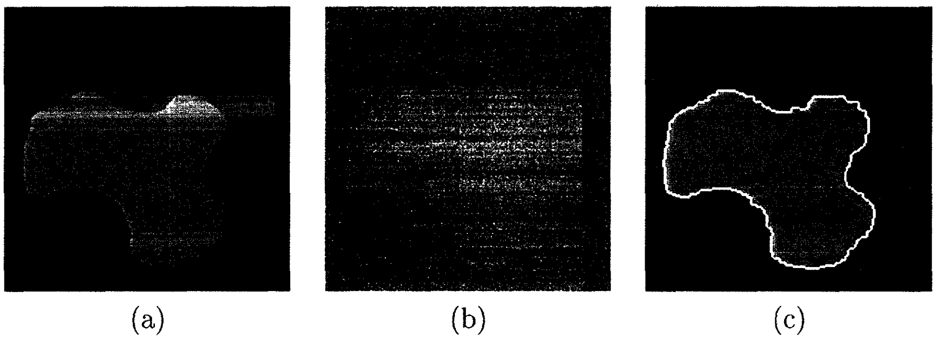


Figure 5.3. (a) A test image; (b) its noisy version (normalized); (c) detected boundary, superimposed onto the noise-free image

the steady state in finite time and has the energy dissipation properties described in Subsection 3.4.2. For usual SIDE force functions, there is no sliding for vector images—just as there was no sliding for scalar images. However, for force functions which are infinite at zero, such as that of Figure 3.8, the sliding property holds, and therefore so does well-posedness. Vector-valued SIDEs are also robust to severe noise, as we show in the experiments.

■ 5.1.1 Experiment 1: Color Images.

We start by applying a vector-valued SIDE to the color image in Figure 5.2. The image in Figure 5.2, (a) consists of two regions: two of its three color channels undergo an abrupt change at the boundary between the regions. More precisely, the {red, green, blue} channel values are $\{0.1, 0.6, 0.9\}$ for the background and $\{0.6, 0.6, 0.5\}$ for the square. Each channel is corrupted with independent white Gaussian noise whose standard deviation is 0.4. The resulting image (normalized in order to make every pixel of every channel be between 0 and 1) is shown in Figure 5.2, (b). We evolve a vector-valued SIDE on the noisy image, until exactly two regions remain. The final boundary,

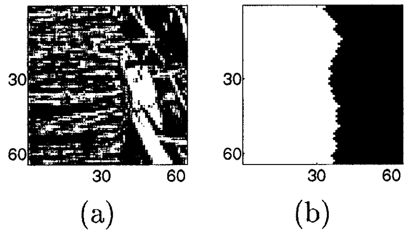


Figure 5.4. (a) Image of two textures: fabric (left) and grass (right); (b) the ideal segmentation of the image in (a).

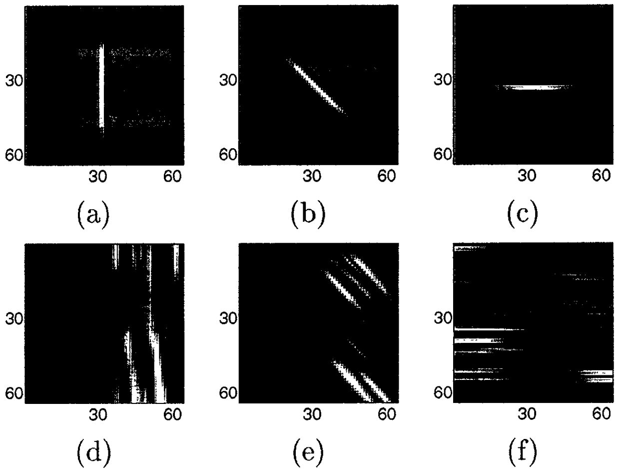


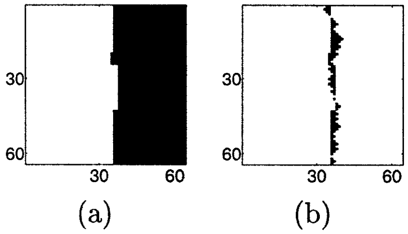
Figure 5.5. (a-c) Filters; (d-f) Filtered versions of the image in Figure 5.4, (a).

superimposed onto the initial image, is depicted in Figure 5.2, (c). Just as in the scalar case, the algorithm is very accurate in locating the boundary: less than 0.2% of the pixels are misclassified in this 100-by-100 image.

A similar experiment, with the same level of noise, is conducted for a more complicated shape, whose image is in Figure 5.3, (a). The result of processing the noisy image of Figure 5.3, (b) is shown in Figure 5.3, (c). In this 200-by-200 image, 0.8% of the pixels are misclassified.

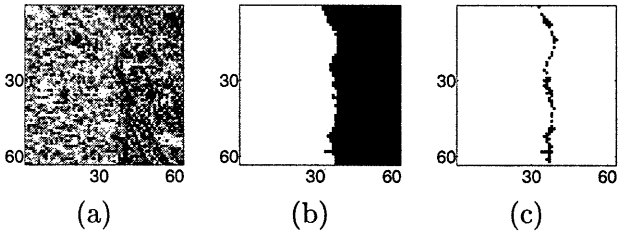
5.1.2 Experiment 2: Texture Images.

Many algorithms for segmenting vector-valued images may be applied to texture segmentation, by extracting features from the texture image and presenting the image consisting of these feature vectors to the segmentation algorithm [36]. This is the paradigm we adopt here. We note that the problem of feature extraction is very important: it is clear that the quality of features will influence the quality of the final segmentation. However, this problem is beyond the scope of the present thesis. The purpose of this experiment is to show the feasibility of texture segmentation using SIDEs, given a



(a) (b)

Figure 5.6. (a) Two-region segmentation, and (b) its deviation from the ideal one.



(a) (b) (c)

Figure 5.7. (a) A different feature image: the direction of the gradient; (b) the corresponding two-region segmentation, and (c) its deviation from the ideal one.

reasonable set of features.

Our first example is the image depicted in Figure 5.4, (a), which consists of two textures: fabric (left) and grass (right). The underlying texture images whose grayscale versions were used to create the image of Figure 5.4, (a), were taken from the VisTex database of the MIT Media Lab [42]. The correct segmentation is in Figure 5.4, (b). We extract features by convolving our image with the three Gabor functions [40] depicted in Figure 5.5, (a-c), which essentially corresponds to computing smoothed directional derivatives. The results of filtering our image with these functions are shown in Figure 5.5, (d-f). These are the three channels of the vector-valued image which we use as the initial condition for the SIDE (5.2). Figure 5.6, (a) is the resulting two-region segmentation, and its difference from the ideal segmentation is in Figure 5.6, (b). The result is very accurate: about 3.2% of the pixels are classified incorrectly, all of which are very close to the initial boundary.

The next example illustrates the importance of the pre-processing step. We again segment the image of Figure 5.4, (a), but this time we use a different feature. Instead of the three features of Figure 5.5, we use a single feature, which is the absolute value of the gradient direction, depicted in Figure 5.7, (a). More precisely, the (i,j)-th pixel value of the feature image is

$$|\text{angle}(u_{i,j+1} - u_{i,j} + \sqrt{-1}(u_{i+1,j} - u_{i,j}))^2|,$$

where $u_{i,j}$ is the (i,j)-th pixel of the original image. This leads to a significant improvement in performance, shown in the rest of Figure 5.7: the shape of the boundary is much closer to that of the ideal boundary, and the number of misclassified pixels is

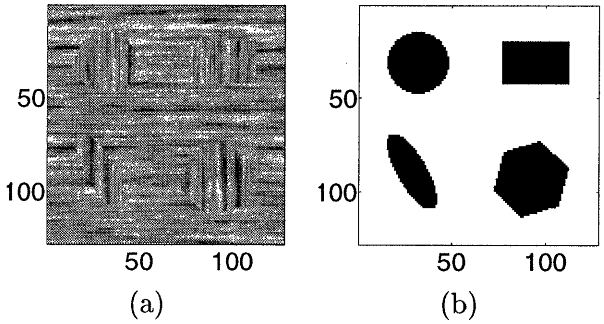


Figure 5.8. (a) Image of two wood textures; (b) the ideal segmentation of the image in (a).

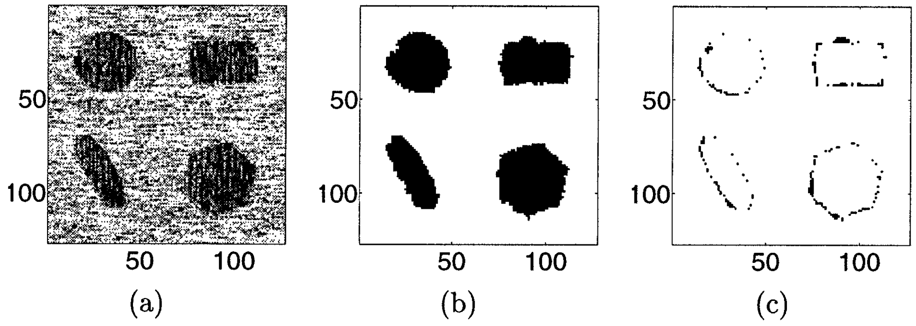


Figure 5.9. (a) Feature image for the wood textures; (b) the corresponding five-region segmentation, and (c) its deviation from the ideal one.

only 2% of the total area of the image.

In our next example, we segment the image of Figure 5.8, which involves two wood textures generated using the Markov random field method of [37]. We again use the direction of the gradient as the sole feature (Figure 5.9, (a)). The five-region segmentation of Figure 5.9, (b) is again very accurate, with the area between our boundary and the ideal one being only 1.6% of the total area of the image (Figure 5.9, (c)).

Our final example of this subsection involves the same wood textures, but arranged in more complicated shapes (Figure 5.10). The feature image and the corresponding segmentation (2.3% of the pixels are misclassified) are shown in Figure 5.11. We will see in the next section that if the two textures only differ in orientation (as in the last two examples), then performance can be improved by using a different version of SIDEs, adapted for processing orientation images.

We close this section by noting that in all the examples of this subsection, our segmentation algorithm performed very well, despite the fact that the feature images were of poor quality. No attempt was made to optimize the pre-processing step of extracting the features.

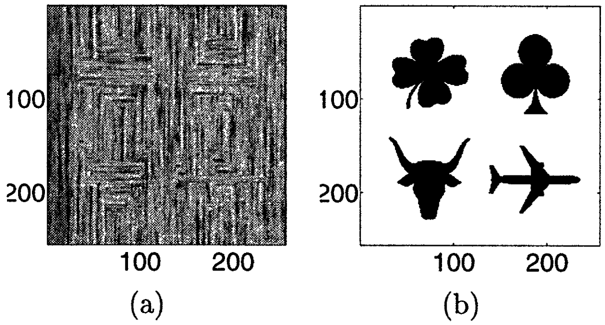


Figure 5.10. (a) Another image of two wood textures; (b) the ideal segmentation of the image in (a).

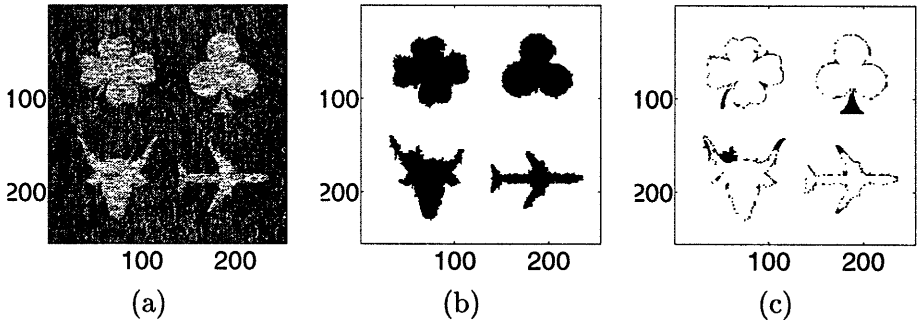


Figure 5.11. (a) Feature image for the wood textures in Figure 5.10, (a); (b) the corresponding five-region segmentation, and (c) its deviation from the ideal one.

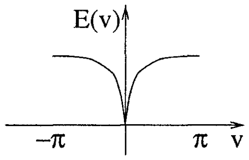


Figure 5.12. A SIDE energy function which is flat at π and $-\pi$ and therefore results in a force function which vanishes at π and $-\pi$.

5.2 Orientation Diffusions.

In [48], Perona used the linear heat diffusion as a basis for developing orientation diffusions. In this section, we follow a similar procedure to adapt SIDEs for processing orientation images. In other words, we consider the case when the image to be processed takes its values on the unit circle $\mathbb{S}^1$, i.e., each pixel value is a complex number with unit magnitude. We define the signed distance $\zeta(a,b)$ between two such numbers a and b as their phase difference:

$$\zeta(a,b) \overset{\text{def}}{=} -\sqrt{-1}\log\frac{a}{b}, \tag{5.3}$$

where log is the principal value, i.e., $-\pi \le \zeta < \pi$.

We would like the force acting between two antipodal points to change continuously as the distance between them changes from $\pi - 0$ to $-\pi + 0$ (i.e., from slightly smaller than π to slightly larger than $-\pi$). We therefore restrict our attention to SIDE energy functions for which $E'(\pi) = E'(-\pi) = 0$, as shown in Figure 5.12. We define the following global energy:

$$\mathcal{E} = \sum_{i,j\text{are neighbors}} E(|\zeta(u_j,u_i)|). \tag{5.4}$$

We define SIDEs for circle-valued images and signals as the following gradient descent equation:

$$\dot{\mathbf{u}} = -\nabla\mathcal{E},$$

with the i-th pixel evolving according to:

$$\dot{u}_i = -\nabla_i\mathcal{E},$$

where ∇_i is the gradient taken on the unit circle $\mathbb{S}^1$. To visualize this evolution, the straight vertical lines of the spring-mass models of Figures 3.1 and 3.3 are replaced with circles: each particle is moving around a circle. After taking the gradients, simplifying, and taking into account merging of pixels, we obtain that the differential equation governing the evolution of the phase angles of u_i's is very similar to the scalar SIDE:

$$\dot{\theta}_i = \frac{1}{m_i}\sum_{j\in A_i} F(\zeta(u_j,u_i))p_{ij}, \tag{5.5}$$

where θ_i is the phase angle of u_i (we use the convention $0 \leq \theta_i < 2\pi$, and identify $\theta_i = 2\pi$ with $\theta_i = 0$). The rest of the notation is the same as in Chapter 3. Two neighboring pixels are merged when they have the same phase angle.

While this evolution has many similarities to its scalar and vector-valued counterparts, it also has important differences, stemming from the fact that it operates on the phase angles. Thus, it is not natural to talk about the mean of the (complex) values of the input image; instead, this evolution preserves the sum of the phases, modulo 2π. This is easily verified by summing up the equations (5.5).

Property 5.1 (Total phase conservation). *The phase angle of the product of all pixel values stays invariant throughout the evolution of (5.5).*

Another important distinction from the scalar and vector SIDEs is the existence of unstable equilibria. Since $F(\pi) = E'(\pi) = 0$, it follows that if two neighboring pixels have values which are two antipodal points on $\mathbb{S}^1$, these points will neither attract not repulse each other. Thus, unlike the scalar and vector-valued equations whose only equilibria were constant images, many other equilibrium states are possible here. The next property, however, guarantees that the only stable equilibria are constant images.

Property 5.2 (Equilibria). *Suppose that all the pixel values of an image* $\mathbf{u}$ *have the same phase. Then* $\mathbf{u}$ *is a stable equilibrium of (5.5). Conversely, such images* $\mathbf{u}$ *are the only stable equilibria of (5.5).*

Proof. If all the phases are the same, then the whole image $\mathbf{u}$ is a single region, which therefore has no neighbors and is not changing. Moreover, if a pixel value is perturbed by a small amount, forces exerted by its neighbors will immediately pull it back. Thus, it is a stable equilibrium.

Suppose now that an equilibrium image $\mathbf{u}$ has more than one region. Let us pick an arbitrary region, call its value u_*, and partition the set of its neighboring regions into two subsets: $U = \{u_1, \ldots, u_p\}$, whose every element pulls u_* in the counter-clockwise direction (i.e., U is comprised of those regions for which the phase of $\frac{u_i}{u_*}$ is positive and strictly less than π), and the set $V = \{v_1, \ldots, v_q\}$, whose elements pull u_* in the clockwise direction (i.e., those regions for which the phase of $\frac{v_i}{u_*}$ is negative and greater than or equal to $-\pi$). One of the sets U, V—but not both—can be empty.

Since our system (5.5) is in equilibrium, it means that the resultant force acting on u_* is zero—i.e., the right-hand side of the corresponding differential equation is zero. Suppose now that u_* is slightly perturbed in the clockwise direction. Since the force function F is monotonically decreasing, this means that the resultant force exerted on u_* by the regions comprising the set V will increase, and the resultant force exerted by U will decrease. The net result will be to further push u_* in the clockwise direction. A similar argument applies if u_* is perturbed in the counter-clockwise direction. Thus, the equilibrium is unstable, which concludes the proof. ■

For any reasonable probabilistic model of the initial data, the probability of attaining an unstable equilibrium during the evolution is zero. In any case, a numerical

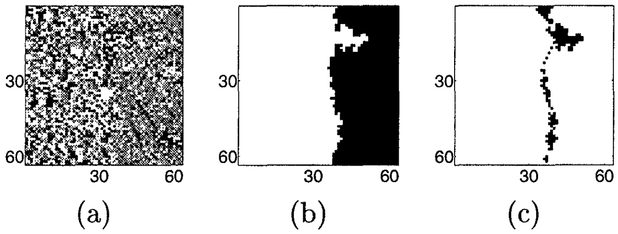


Figure 5.13. (a) The orientation image for Figure 5.7, (a); (b) the corresponding two-region segmentation, and (c) its deviation from the ideal one.

implementation can be designed to avoid such equilibria. Therefore, in the generic case, the steady state of the evolution is a stable equilibrium, which, by the above property, is a constant image. This corresponds to the coarsest segmentation (everything is one region), just like in the scalar-valued and vector-valued cases.

Since there is no notion of a maximum on the unit circle, there is no notion of a "maximum principle", either. This means, moreover, that we cannot mimic the proof of Property 3.2 (finite evolution time) of the scalar evolutions. This property does hold for the evolutions on a circle, but the proof is different. Specifically, Property 3.7 holds here, with a similar proof, which means that between two consecutive mergings, the global energy is a concave decreasing function of time (Figure 3.7). It will therefore reach zero in finite time, at which point the evolution will be at one of its equilibria.

Property 5.3 (Finite evolution time). *The SIDE (5.5) reaches its equilibrium in finite time, starting with any initial condition.*

■ 5.2.1 Experiments.

To illustrate segmentation of orientation images, we use the same texture images which we used in the previous section. To extract the orientations, we use the direction of the gradient. The (i,j)-th pixel value of the orientation image is

$$\text{angle}[(u_{i,j+1} - u_{i,j} + \sqrt{-1}(u_{i+1,j} - u_{i,j}))]^2,$$

where $u_{i,j}$ is the (i,j)-th pixel value of the raw texture image. (Note that the absolute value of this orientation image was used as a feature image in the previous section.) The expression in the above formula is squared so as to equate the phases which differ by π.

The orientations for the fabric-and-grass image are shown in Figure 5.13, (a). We present this image as the initial condition to the circle-valued SIDE (5.5) and evolve it until two regions remain. The resulting segmentation is depicted in Figure 5.13, (b), and its difference from the ideal one is in 5.13, (c). About 4.3% of the total number of pixels are classified incorrectly. It is not surprising that this method performs slightly worse on this example than the evolutions of the previous section. Indeed, if a human were asked to segment the original texture image based purely on orientation, he might

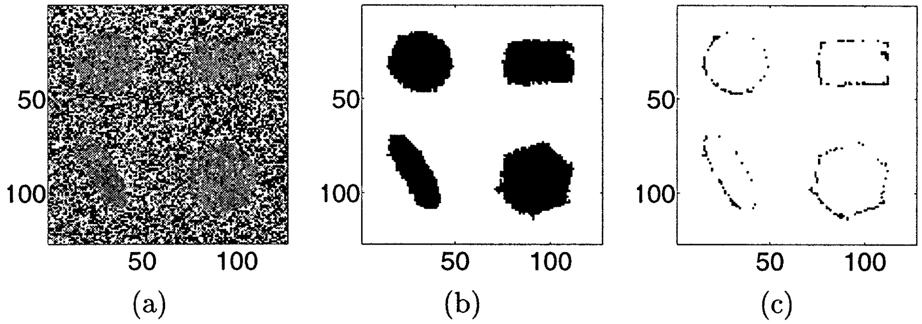


Figure 5.14. (a) The orientation image for Figure 5.9, (a); (b) the corresponding five-region segmentation, and (c) its deviation from the ideal one.

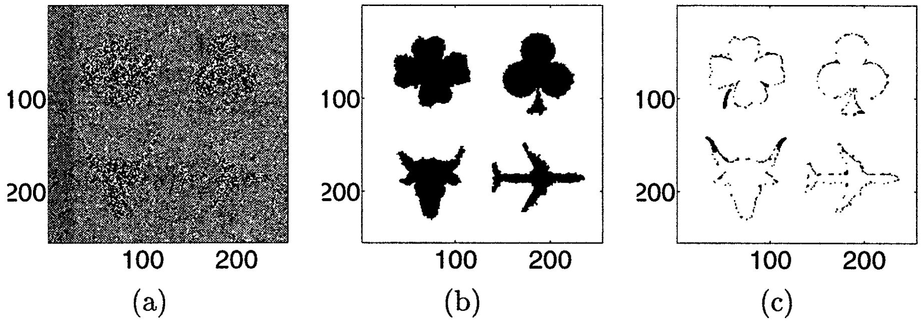


Figure 5.15. (a) The orientation image for Figure 5.10, (a); (b) the corresponding five-region segmentation, and (c) its deviation from the ideal one.

make similar errors. Note in particular that the protrusion in the upper portion of the boundary found by the SIDE corresponds to a horizontally oriented piece of grass in the original image, which can be mistaken for a portion of the fabric.

In the second example, however, the orientation information is very appropriate for characterizing and discriminating the two differently oriented wood textures. The orientation image is in Figure 5.14, (a). The resulting five-region segmentation of Figure 5.13, (b) incorrectly classifies only 252 pixels (1.5%) in the 128-by-128 image, which is 14 pixels better than the method of the previous section. A more dramatic improvement is achieved in the example of Figure 5.15, which shows the five-region segmentation of the image in Figure 5.10, (a), using the circle-valued SIDE (5.5). Only 1.5% of the pixels are misclassified, as compared to 2.3% using the method of the previous section.

5.3 Conclusion.

In this chapter, we generalized SIDEs to the situations when the image to be processed is not scalar-valued. We described the properties of the resulting evolutions and demonstrated their application to segmenting color, texture, and orientation images.

Chapter 6

Conclusions and Future Research

6.1 Contributions of the Thesis.

IN this thesis we have presented a new approach to edge enhancement and segmentation which is robust to very high levels of noise and blurring. Our approach is based on a new class of evolution equations for the processing of imagery and signals which we have termed stabilized inverse diffusion equations or SIDEs. These evolutions, which have discontinuous right-hand sides, have conceptual and mathematical links to other evolution-based and variational methods in signal and image processing, such as the Perona-Malik equation, shock filters, total variation minimization, region merging methods, the Geman-Reynolds functional, and the Mumford-Shah functional. SIDEs, however, have their own unique qualitative characteristics and properties which were described in Chapter 3 and led to improved performance. Specifically, a number of stability properties were derived, such as well-posedness of the 1-D equation, the maximum principle, and—for almost all input images—the invariance of the resulting segmentation with respect to small changes in the input image. SIDEs were further characterized as gradient descent equations for certain energy functionals. Several classes of Lyapunov functionals were found for these equations.

SIDEs' superior performance was demonstrated in Chapter 3 through a multitude of examples, both in 1-D and in 2-D. Their robustness—in contrast with other methods—was shown in segmenting signals with heavy-tailed noise and images with high levels of blurring and speckle noise, such as ultrasound and SAR images.

Chapter 4 provided an extensive performance analysis and a probabilistic interpretation of a variant of SIDEs. We showed that it produced maximum likelihood solutions to certain binary classification/edge detection problems in 1-D. We conjectured that similar results hold in 2-D. Extensive performance analysis for the 1-D problem of finding one change in mean was done, producing bounds on the probabilities of errors. For this problem, it was also proved that the algorithm is optimal with respect to a certain H_∞-like criterion. The experiments in both 1-D and 2-D showed that the algorithm performs extremely well, and is robust to large-amplitude noise and blurring.

We also presented a fast implementation and compared it with the dynamic programming solution of the 1-D problem. Dynamic programming was faster in certain instances, but took up more memory and did not generalize to 2-D. We explored the links between our non-linear diffusion equation and the Mumford-Shah variational method of image segmentation, and showed that a certain particular case of these is a linear programming problem.

Thus, the contribution of Chapter 4 was two-fold. First, we presented a fast and robust 1-D edge detection and 2-D image segmentation method. Second, we established a link between deterministic methods for image restoration and segmentation (based on non-linear diffusions and variational formulations) and a probabilistic framework. This leads to a deeper understanding of these methods: both of their performance, and of how to use them in a variety of situations (e.g., in Section 4.4 this meant pre-processing the data by forming the log-likelihood ratios). As we will argue in the next section, we have no doubt that these lines of investigation can and should be pursued further.

Finally, in Section 5 we demonstrated that our framework can be easily adapted to non-scalar-valued images. Specifically, we used the result from Chapter 3 which showed that a scalar-valued SIDE is the steepest descent equation for a certain energy functional. We then generalized SIDEs by deriving the steepest descent equations for similar energy functionals in vector-valued and circle-valued cases. We showed that many properties of the scalar-valued SIDEs applied, and pointed out several important differences. We successfully applied the resulting evolution to the segmentation of color, texture, and orientation images.

■ 6.2 Future Research.

In this section, we list several interesting problems motivated by this thesis.

■ 6.2.1 Feature Extraction.

Once one admits the concept of sliding surfaces for signal or image evolution, the question immediately arises as to whether one can design sliding surfaces other than those used in this work. In particular, the sliding surfaces reflecting the enforced equality of neighboring points or pixels, directly correspond to piecewise-constant approximations of signals, and the resulting SIDE evolution in essence produces an adapted sequence of staircase approximations to a signal or an image. An open question is whether it is also possible to produce a sequence of other approximations (e.g., linear splines) to signals and images by appropriately defining the sliding surfaces. The challenge is to design these surfaces in such a way that the resulting SIDE evolution is both fast and robust.

■ 6.2.2 Choosing the Force Function.

We mentioned in Chapter 3 that choosing a SIDE force function best suited for a particular application is an open research question. We also partially addressed this issue in Chapter 4, by describing and analyzing the problems for which $F(v) = \text{sgn}(v)$ was the optimal choice. This choice was optimal in the sense that it resulted in the maximum likelihood solutions to change detection problems for certain probabilistic

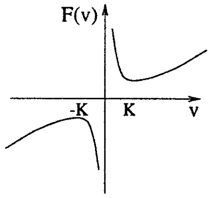


Figure 6.1. A force function which results in faster removal of outliers.

models of the input signal. It is unclear how to choose force functions for other signal models; an interesting question is for what models this can be done so as to guarantee that the solution produced by the SIDE is the maximum likelihood solution.

Another question is that of robustness properties of the SIDEs corresponding to different shapes of the force function. Intuitively, if the goal is segmentation in the presence of outliers, then it is appropriate to diffuse quickly both in the areas of very large gradient (corresponding to the outliers), and in the areas of very small gradient (corresponding to small-amplitude noise). Ideally, the minimum diffusion speed would be at the locations with intermediate values of the gradient, corresponding to edges. We are then led to the form of a force function depicted in Figure 6.1, which is the inverse of a Perona-Malik force function, and has a unique minimum at some location K. If the parameters of the probabilistic model of the input signal (such as the standard deviation of the noise, the magnitude and frequency of the outliers) are fixed, then it is natural to expect that some value of K is, in some sense, optimal. Uncovering this relationship between the model and the corresponding value of the parameter K is an interesting research topic.

■ 6.2.3 Choosing the Stopping Rule.

Another interesting question is how to stop a SIDE so as to obtain the best segmentation. When the force function is $F(v) = \text{sgn}(v)$ and the number of regions is an exponential random variable, the answer to this question is given by Proposition 4.6 of Chapter 4. We have been unable, however, to answer this question for the general form of the SIDE force function F. In our examples, we assumed that the desired number of regions was known, and stopped the evolution when that number was attained. In some applications where the number of objects of interest is not known a priori, this may not be a realistic assumption.

As we remarked in Chapter 4, one way to address the question of when to stop is by looking at the size of the energy of the region being removed, or at the time required to remove it. If a region has large energy, or if a long time is required to remove it, it is more likely to be significant, and less likely to be due to noise.

6.2.4 Image Restoration.

As mentioned in Chapter 3, while SIDEs may do an excellent job of segmentation and location of edges, they do not directly provide good estimates of the values within regions or between edges. This limitation can be addressed, for example, by using a SIDE for segmentation and then using optimal linear estimation or filtering within each segmented region in order to get both accurate edge estimates and denoising within each so-identified region.

6.2.5 PDE Formulation.

The problem of re-casting SIDEs as PDEs similar to (3.27) of Chapter 3 is of great theoretical interest, because PDEs have been the unifying language of most of the researchers in the field of non-linear scale spaces and geometry-driven diffusion. A PDE formulation is also important because it will allow one to obtain certain results concerning the evolution. For example, the analysis of the invariance of the equation to various coordinate transformations of the image domain is clearly more convenient to carry out in the continuous-space setting.

6.2.6 Further Probabilistic Analysis.

An important topic for future investigations is whether results similar to those of Chapter 4 can be obtained for signals whose samples come from more than two probability distributions. In addition, we would like to generalize these results to the situation when the samples of the observed process are not conditionally independent. This would make possible the design of probabilistic texture segmentation algorithms. We also would like to explore the case of unknown probability densities for the binary problem, which means devising ways of learning the log-likelihood ratios from the data. Proving the conjectures of Section 4.7 also has high priority.

The most ambitious question is perhaps analyzing the performance in 2-D, which amounts to analyzing random fields with abrupt changes. In order to understand the sensitivity of a segmentation algorithm to noise, we need to understand how the edges produced by the algorithm will change if the input image is perturbed by a spatial random field.

6.2.7 Prior Knowledge.

Another avenue of future research is understanding how to incorporate prior knowledge, which is a very important question not only for segmentation, but for any other image processing task. For example, we saw in Chapter 4 that SIDEs can be modified to make use of a prior distribution on the number of desired regions in the segmentation. It remains to be seen, however, how to incorporate other knowledge, for instance, information about sizes and shapes of the regions.

Appendix A

Proof of Lemma on Sliding (Chapter 3)

To simplify notation, we replace n_i with i in (3.11) and re-write the system in terms of $v_i = u_{i+1} - u_i$:

$$\begin{aligned} \dot{v}_i &= \frac{1}{m_{i+1}}(F(v_{i+1}) - F(v_i)) - \frac{1}{m_i}(F(v_i) - F(v_{i-1})) \\ i &= 1, \ldots, p-1. \end{aligned} \tag{A.1}$$

We need to prove that if $(i_1, \ldots, i_{p-1})$ is any permutation of $(1, \ldots, p-1)$, then, as $\mathbf{v}$ approaches $S = \cap_{k=1}^{m} S_{i_k} \setminus (\cup_{k=m+1}^{p-1} S_{i_k})$, $\lim(\dot{v}_{i_q} \mathrm{sign}(v_{i_q})) \leq 0$ for all integers q between 1 and m, and for at least one such q the inequality is strict (i.e., the trajectories enter S transversally). Note that for every point $s \in S$ and every quadrant Q, we only need to find one sequence of $\mathbf{v}$'s approaching s from Q and satisfying these inequalities. This is because the solutions vary continuously inside each quadrant.

Fix $v_{i_{m+1}}, \ldots, v_{i_{p-1}}$ at non-zero values, let

$$\varepsilon = \frac{1}{2} \min_{m+1 \leq j \leq p-1} |v_{i_j}|,$$

let initially $\delta = \varepsilon$, set $|v_{i_1}| = \ldots = |v_{i_m}| = \delta$, and drive $\mathbf{v}$ towards S by letting δ go to zero. Take an arbitrary index q between 1 and m. By our construction, v_{i_q} is approaching zero, and either $v_{i_q} = \delta > 0$ or $v_{i_q} = -\delta < 0$. If $v_{i_q} = \delta$, then, by construction, $v_{i_q} \leq |v_{i_{q \pm 1}}|$, implying $F(v_{i_q}) \geq F(v_{i_{q \pm 1}})$, which makes the RHS of (A.1) for $i = i_q$ non-positive: $\lim_{\delta \to 0} \dot{v}_{i_q} \leq 0$. If $m < p - 1$, then there is a j between 1 and m such that at least one of the two neighbors of v_{i_j} is in the set $\{v_{i_{m+1}}, \ldots, v_{i_{p-1}}\}$, and whose absolute value is therefore staying above ε: $|v_{i_{j+1}}| > \varepsilon$ or $|v_{i_{j-1}}| > \varepsilon$. Without loss of generality, suppose it is the left neighbor: $|v_{i_{j-1}}| > \varepsilon$. If $m = p - 1$, define $j = 1$. If our arbitrary q happens to be equal to this j, then

$$F(v_{i_q}) - F(v_{i_{q-1}}) = F(v_{i_j}) - F(v_{i_{j-1}}) > F(v_{i_j}) - F(\varepsilon),$$

and hence (A.1) for $i = i_j$ has a strictly negative limit: $\lim_{\delta \to 0} \dot{v}_{i_j} < 0$. Similar reasoning for the case $v_{i_q} = -\delta$ leads to $\lim_{\delta \to 0} \dot{v}_{i_q} \geq 0$, and, if it happens that $q = j$, then $\lim_{\delta \to 0} \dot{v}_{i_q} > 0$. ■

Appendix B

Proofs for Chapter 4.

B.1 Proof of Proposition 4.1.

Let $\mathbf{h} = \mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$, and suppose that $\mathbf{h}$ has an upward edge at location i: $h_i = 0$, and $h_{i+1} = 1$. Suppose further that there is no upward α-crossing at i. We only treat the case when the signal values are not equal to the threshold: $u_i^0 \neq \alpha$, $u_{i+1}^0 \neq \alpha$. The degenerate cases when one or more signal values are equal to the threshold are handled similarly.

Consider two possibilities.

Case 1. $u_i^0 > \alpha$ (Figure B.1, (b)).

Changing h_i from 0 to 1 will either maintain the same number of edges (if $i > 1$ and $h_{i-1} = 0$), or reduce the number of edges by one (if $i = 1$), or reduce the number of edges by two (if $i > 1$ and $h_{i-1} = 1$). It will also increase $\phi(\mathbf{u}^0, \mathbf{h})$ (by the amount $u_i^0 - \alpha$), which contradicts the assumption that $\mathbf{h}$ is the best hypothesis with ν or fewer edges. Thus, u_i^0 has to be

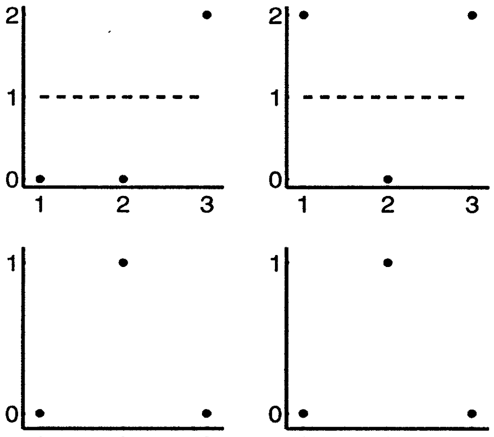


Figure B.1. Samples of a signal (top plots) and impossible edge configurations of optimal hypotheses (bottom plots).

smaller than α.

Case 2. $u_i^0 < \alpha$ and $u_{i+1}^0 < \alpha$ (Figure B.1, (a)).
Changing h_{i+1} from 1 to 0 will either maintain the same number of edges (if $i < N-1$ and $h_{i+2} = 1$), or reduce the number of edges by one (if $i = N-1$), or reduce the number of edges by two (if $i < N-1$ and $h_{i+2} = 0$). It will also increase $\phi(\mathbf{u}^0, \mathbf{h})$ (by the amount $\alpha - u_{i+1}^0$), again violating the assumption that $\mathbf{h}$ is the best hypothesis with ν or fewer edges. Combining this with Case 1, we see that $\mathbf{u}$ must have an upward α-crossing at i.

It is similarly shown that every downward edge of $\mathbf{h}$ occurs at a downward α-crossing of $\mathbf{u}^0$. ■

■ B.2 Proof of Proposition 4.2: SIDE as a Maximizer of a Statistic.

We note that this proof relies on Lemma 4.1, proved in Subsection 4.3.1.

We first show that at any time t between 0 and t_f, the rate of reduction of $\phi(\mathbf{u}(t), \mathbf{h})$ is the largest for $\mathbf{h} = \mathbf{h}^*(\mathbf{u}(t_f))$, among all hypotheses $\mathbf{h}$ with $\nu_\alpha(t_f)$ or fewer edges. To simplify notation, define $\nu = \nu_\alpha(t_f)$, $g_{j+1} = N$, and let $g_1, \ldots, g_j$ be the edge locations (from left to right) of an arbitrary hypothesis $\mathbf{h}$ with ν or fewer edges (i.e., $j \le \nu$). Without loss of generality, assume that the leftmost edge is upward, so that $h_{g_1} = 0$. Then

$$
\begin{aligned}
\dot{\phi}(\mathbf{u}(t), \mathbf{h}) &= \mathbf{h}^T \dot{\mathbf{u}}(t) && \text{(B.1)}\\
&= \sum_{i \text{ is odd}} \sum_{n=g_i+1}^{g_{i+1}} \dot{u}_n(t) && \text{(B.2)}\\
&= \sum_{i \text{ is odd}} \left\{ \sum_{n=g_i+1}^{N} \dot{u}_n(t) - \sum_{n=g_{i+1}+1}^{N} \dot{u}_n(t) \right\}, && \text{(B.3)}
\end{aligned}
$$

Note that, for any integer s such that $1 \le s \le N-1$ and $u_{s+1}(t) - u_s(t) \ne 0$, we have, by summing up Equations (4.1) from $n = s+1$ to $n = N$,

$$\sum_{s+1}^{N} \dot{u}_n(t) = -\operatorname{sgn}(u_{s+1}(t) - u_s(t)) = \pm 1. \qquad \text{(B.4)}$$

If $u_{s+1}(t) = u_s(t)$, let p be the smallest index to the left of s such that $u_{p+1}(t) = u_s(t)$, and let q be the largest index to the right of s such that $u_q(t) = u_s(t)$. If $q \le N-1$

and $p \geq 1$, then, according to Equations (4.1) and (4.3), we have:

$$\begin{aligned}\sum_{n=s+1}^{N} \dot{u}_n(t) &= \sum_{n=s+1}^{q} \dot{u}_n(t) + \sum_{n=q+1}^{N} \dot{u}_n(t) \\ &= (q-s)\dot{u}_s - \operatorname{sgn}(u_{q+1}(t) - u_q(t)) \\ &= (q-s)\frac{1}{q-p}(\operatorname{sgn}(u_{q+1}(t) - u_q(t)) - \operatorname{sgn}(u_{p+1}(t) - u_p(t))) \\ &\quad -\operatorname{sgn}(u_{q+1}(t) - u_q(t)) \\ &= -\frac{1}{q-p}\operatorname{sgn}(u_{q+1}(t) - u_q(t)) \\ &\quad -\frac{1}{q-p}\operatorname{sgn}(u_{p+1}(t) - u_p(t)) \geq -1, \end{aligned} \tag{B.5}$$

since $q - p \geq 2$. The same inequality is obtained if $q = N$ or $p = 0$. Combining (B.5) and (B.4), we see that $\sum_{n=s+1}^{N} \dot{u}_n(t) \geq -1$ for any s. Therefore, the minimal possible value for (B.3) is (-1) times the number of sums of the form $\sum_{n=s+1}^{N} \dot{u}_n(t)$ in that expression. This is $-j$, which is, by assumption, not smaller than $-\nu$:

$$\dot{\phi}(\mathbf{u}(t), \mathbf{h}) \geq -j \geq -\nu.$$

Now note that in this double inequality, both equalities are achieved for all time t, $0 \leq t \leq t_f$, when $\mathbf{h} = \mathbf{h}^*(\mathbf{u}(t_f))$. Indeed, since in this case $j = \nu$, and—as easily seen from the definition of ϕ—$g_1, \ldots, g_\nu$ are α-crossings of $\mathbf{u}(t_f)$ (and, therefore—by Lemma 4.1—also of $\mathbf{u}(t)$ for $0 \leq t \leq t_f$), we have that for $t \in [0, t_f]$,

$$-\operatorname{sgn}(u_{g_i+1}(t) - u_{g_i}(t)) = \begin{cases} -1 & \text{if } i \text{ is odd} \\ 1 & \text{if } i \text{ is even.} \end{cases}$$

Inserting this into (B.4), and then back into (B.3), we see that in this case, (B.3) is equal to $-\nu$. By definition of $\mathbf{h}^*(\mathbf{u}(t_f))$, $\phi(\mathbf{u}(t_f), \mathbf{h})$ is the largest for $\mathbf{h} = \mathbf{h}^*(\mathbf{u}(t_f))$. On the other hand, we just showed that the amount of the reduction of $\phi(\mathbf{u}(t), \mathbf{h})$ during the evolution was the greatest for $\mathbf{h} = \mathbf{h}^*(\mathbf{u}(t_f))$, over all possible hypotheses with ν or fewer edges. Therefore, $\phi(\mathbf{u}(0), \mathbf{h})$ must also have been the largest for $\mathbf{h} = \mathbf{h}^*(\mathbf{u}(t_f))$, over the same set of hypotheses, which is the statement of the Proposition. ■

The statement of the same proposition in [51] is as follows.

Proposition B.1. *Fix the initial condition* $\mathbf{u}^0$ *of the SIDE (4.1), and let* $\mathbf{u}(t)$ *be the corresponding solution. Suppose that a statistic* ϕ *satisfies two conditions:*

1) $\frac{d}{dt}\left\{\phi(\mathbf{u}(t), \mathbf{h}) - \mathbf{h}^T\mathbf{u}(t))\right\} = 0$;

2) *there exists* $\alpha \in I\!R$ *such that,* $\forall t \geq 0$*, the optimal hypothesis* $\mathbf{h}^*(\mathbf{u}(t))$ *is generated by the set of all* α*-crossings of* $\mathbf{u}(t)$*.*

Let $\nu_\alpha(t)$ be the number of α-crossings of $\mathbf{u}(t)$. Then

$$\mathbf{h}^*_{\le\nu_\alpha(t)}(\mathbf{u}^0) = \mathbf{h}^*(\mathbf{u}(t)). \qquad \blacksquare$$

We now show that the two statements are equivalent.

Proposition B.2. *Propositions 4.2 and B.1 are equivalent, in the following sense.*
i) If $\phi(\mathbf{u},\mathbf{h})$ is as in Proposition 4.2, it satisfies the two conditions of Proposition B.1.
ii) Suppose that $\phi'(\mathbf{u},\mathbf{h})$ satisfies the two conditions of Proposition B.1 for any initial data $\mathbf{u}^0 \in I\!R^N$ (where the constant α may depend on $\mathbf{u}^0$), and suppose that $\phi(\mathbf{u},\mathbf{h})$ is as in Proposition 4.2. Then, for all $\mathbf{u} \in I\!R^N$ and for all $\mathbf{h} \in \{0,1\}^N \backslash \{\mathbf{0},\mathbf{1}\}$ (where $\mathbf{1} = (1,\ldots,1)^T \in I\!R^N$), $\phi(\mathbf{u},\mathbf{h})$ and $\phi'(\mathbf{u},\mathbf{h})$ can only differ by a function of $\sum_{i=1}^N u_i$:

$$\phi'(\mathbf{u},\mathbf{h}) - \phi(\mathbf{u},\mathbf{h}) = f\left(\sum_{i=1}^N u_i\right),$$

for some function $f : I\!R \to I\!R$, and thus the optimal hypotheses with respect to ϕ and ϕ' are the same.

Lemma B.1. *Let $\mathbf{u}(t)$ be the solution of the SIDE (4.1). Suppose that a function $\psi : I\!R^N \to I\!R$ satisfies*

$$\frac{d}{dt}(\psi(\mathbf{u}(t))) = 0, \tag{B.6}$$

for any initial data $\mathbf{u}^0$. Then ψ only depends on the sum of the entries of its argument—i.e., there is a function $f : I\!R \to I\!R$ such that

$$\psi(\mathbf{u}) = f\left(\sum_{i=1}^N u_i\right).$$

Proof. We first show that the partial derivatives of ψ with respect to all the variables are equal to each other, using the identity

$$\frac{d}{dt}(\psi(\mathbf{u}(t))) = \sum_{i=1}^N \frac{\partial\psi}{\partial u_i}\dot{u}_i. \tag{B.7}$$

Take an initial condition for which $u_1^0 < u_2^0 < \ldots < u_{N-1}^0 < u_N^0$. It then follows from (4.1) that $\dot{u}_1 = 1$, $\dot{u}_N = -1$, and $\dot{u}_i = 0$ for $2 \le i \le N-1$. Substituting these into (B.7) and using (B.6), we get:

$$\frac{\partial\psi}{\partial u_1} = \frac{\partial\psi}{\partial u_N}. \tag{B.8}$$

Now take an initial condition for which $u_1^0 < u_2^0 < \ldots < u_{N-1}^0$ and $u_N^0 < u_{N-1}^0$. Then $\dot{u}_1 = \dot{u}_N = 1$, $\dot{u}_{N-1} = -2$, and $\dot{u}_i = 0$ for $2 \leq i \leq N-2$, and therefore

$$\frac{\partial \psi}{\partial u_1} - 2\frac{\partial \psi}{\partial u_{N-1}} + \frac{\partial \psi}{\partial u_N} = 0,$$

which, combined with (B.8), gives:

$$\frac{\partial \psi}{\partial u_1} = \frac{\partial \psi}{\partial u_{N-1}}.$$

Proceeding inductively in a similar fashion, we obtain:

$$\frac{\partial \psi}{\partial u_1} = \frac{\partial \psi}{\partial u_2} = \ldots = \frac{\partial \psi}{\partial u_N}. \tag{B.9}$$

Let

$$\begin{aligned} w_i &= u_{i+1} - u_i \text{ for } i = 1, \ldots, N-1, \\ w_N &= \sum_{i=1}^{N} u_i. \end{aligned}$$

Then it is easily verified by direct substitution that

$$\begin{aligned} u_k &= \frac{1}{N}\left(\sum_{i=1}^{N-1} i w_i + w_N\right) - \sum_{i=k}^{N-1} w_i, \text{ for } k = 1, \ldots, N-1 \\ u_N &= \frac{1}{N}\left(\sum_{i=1}^{N-1} i w_i + w_N\right), \end{aligned}$$

and therefore

$$\frac{\partial u_k}{\partial w_i} = \begin{cases} \frac{i}{N}, & i < k \\ \frac{i}{N} - 1, & k \leq i < N-1 \\ \frac{1}{N}, & i = N. \end{cases}$$

This means that, if, $i \neq N$,

$$\begin{aligned} \frac{\partial}{\partial w_i}\{\psi(\mathbf{u}(\mathbf{w}))\} &= \sum_{k=1}^{N} \frac{\partial \psi}{\partial u_k}\frac{\partial u_k}{\partial w_i} \\ &= \frac{\partial \psi}{\partial u_k}\left[\sum_{k=1}^{i}(\frac{i}{N} - 1) + \sum_{k=i+1}^{N} \frac{i}{N}\right] \\ &= \frac{\partial \psi}{\partial u_k}\left[\frac{i^2}{N} - i + (N-i)\frac{1}{N}\right] = 0, \end{aligned}$$

where we used the fact (B.9) that $\frac{\partial\psi}{\partial u_k}$ is the same for all k. If $i = N$, then

$$\begin{aligned}\frac{\partial}{\partial w_i}\{\psi(\mathbf{u}(\mathbf{w}))\} &= \sum_{k=1}^{N}\frac{\partial\psi}{\partial u_k}\frac{\partial u_k}{\partial w_N} \\ &= \frac{\partial\psi}{\partial u_k}\left[\sum_{k=1}^{N}\frac{1}{N}\right] \\ &= \frac{\partial\psi}{\partial u_k}.\end{aligned}$$

Thus, ψ does not depend on $w_1, \dots, w_{N-1}$, only on $w_N = \sum_{i=1}^{N} u_i$. ■

Proof of Proposition B.2.

(i) Is straightforward.

(ii) According to Lemma B.1 above, if $\phi'(\mathbf{u}, \mathbf{h})$ satisfies Condition 1 of Proposition B.1, then

$$\phi'(\mathbf{u}, \mathbf{h}) - \mathbf{h}^T\mathbf{u}$$

is a function of $\mathbf{h}$ and $\sum_{i=1}^{N} u_i$ only. It also follows from the same Lemma that α of Proposition B.1 may only depend on $\sum_{i=1}^{N} u_i$. Therefore, there is a function ψ such that

$$\phi'(\mathbf{u}, \mathbf{h}) = \mathbf{h}^T(\mathbf{u} - \mathbf{a}) + \psi(\mathbf{h}, \sum_{i=1}^{N} u_i).$$

Suppose that ψ depends on $\mathbf{h}$ for $\mathbf{h} \neq \mathbf{0}, \mathbf{1}$. We presently show that this would lead to violating Condition 2 of Proposition B.1. Take $\mathbf{h}_1, \mathbf{h}_2 \in \{0,1\}^N \backslash \{\mathbf{0}, \mathbf{1}\}$, and $S \in \mathbb{R}$, such that

$$\psi(\mathbf{h}_1, S) > \psi(\mathbf{h}_2, S). \tag{B.10}$$

Denote the number of samples where $\mathbf{h}_1$ and $\mathbf{h}_2$ are different by p:

$$p = \|\mathbf{h}_2 - \mathbf{h}_1\|_1 \geq 1.$$

Let k be the number of zeros in $\mathbf{h}_2$:

$$k = N - \|\mathbf{h}_2\|_1,$$

and let

$$\varepsilon = \frac{\psi(\mathbf{h}_1, S) - \psi(\mathbf{h}_2, S)}{2p} > 0.$$

Case 1. *There are i and j such that $h_{1,i} = h_{2,i} = 1$ and $h_{1,j} = h_{2,j} = 0$.*

Let $u_m = \alpha - \varepsilon$ if $h_{2,m} = 0$ and $m \neq j$, and let $u_m = \alpha + \varepsilon$ if $h_{2,m} = 1$ and $m \neq i$. If

$S - N\alpha - (N-2k)\varepsilon \geq 0$, let $u_i = \alpha + \varepsilon + S - N\alpha - (N-2k)\varepsilon$, and let $u_j = \alpha - \varepsilon$. Otherwise, let $u_i = \alpha + \varepsilon$ and $u_j = \alpha - \varepsilon + S - N\alpha - (N-2k)\varepsilon$. Then

$$\begin{aligned}\phi'(\mathbf{u}, \mathbf{h}_1) - \phi'(\mathbf{u}, \mathbf{h}_2) &= -p\varepsilon + \psi(\mathbf{h}_1, S) - \psi(\mathbf{h}_2, S) \\ &= -p\varepsilon + 2p\varepsilon = p\varepsilon > 0. \end{aligned} \tag{B.11}$$

On the other hand, the edges of $\mathbf{h}_2$ coincide with the α-crossings of $\mathbf{u}$: $u_m > \alpha$ whenever $h_{2,m} = 1$ and $u_m < \alpha$ whenever $h_{2,m} = 0$. Therefore, if ϕ' is to satisfy Condition 2 of Proposition B.1, $\mathbf{h}_2$ has to be the optimal hypothesis for $\mathbf{u}$, which contradicts (B.11). We also note that $\sum_{m=1}^{N} u_m = S$ by construction.
Case 2. *There is no i for which $h_{1,i} = h_{2,i} = 1$, and there is no j for which $h_{1,j} = h_{2,j} = 0$, i.e., $h_{1,i} = 1 - h_{2,i}$ for $i = 1, \ldots, N$.*
If $2 \leq k \leq N-2$, let i and j be such that $h_{2,i} = h_{2,j} = 1$. Let $\mathbf{h}_3$ be obtained from $\mathbf{h}_2$ by changing the i-th entry from zero to one and the j-th entry from one to zero. Then either $\psi(\mathbf{h}_1, S) \neq \psi(\mathbf{h}_3, S)$, or $\psi(\mathbf{h}_2, S) \neq \psi(\mathbf{h}_3, S)$ (or both), and both pairs $(\mathbf{h}_1, \mathbf{h}_3)$ and $(\mathbf{h}_2, \mathbf{h}_3)$ fall under Case 1 considered above.

If $k = 1$, let i be the index for which $h_{2,i} = 0$. Without loss of generality, assume that $i \neq 1$ and $i \neq N$. Form $\mathbf{h}'_1$ and $\mathbf{h}'_2$ as follows:

$$\begin{aligned} h'_{2,m} &= h_{2,m} \text{ for } m = 1, \ldots, N-1 \\ h'_{2,N} &= 0 \\ h'_{1,m} &= h_{1,m} \text{ for } m = 2, \ldots, N \\ h'_{1,1} &= 1. \end{aligned}$$

In other words,

$$\begin{aligned} h_1 &= (0, 0, \ldots, 0, 1, 0, \ldots, 0, 0)^T, \\ h'_1 &= (1, 0, \ldots, 0, 1, 0, \ldots, 0, 0)^T, \\ h_2 &= (1, 1, \ldots, 1, 0, 1, \ldots, 1, 1)^T, \\ h'_2 &= (1, 1, \ldots, 1, 0, 1, \ldots, 1, 0)^T. \end{aligned}$$

Then one of the following holds:

$$\begin{aligned} \psi(\mathbf{h}_1, S) &\neq \psi(\mathbf{h}'_1, S), \\ \psi(\mathbf{h}_2, S) &\neq \psi(\mathbf{h}'_2, S), \\ \psi(\mathbf{h}'_1, S) &\neq \psi(\mathbf{h}'_2, S), \end{aligned}$$

and all three pairs $(\mathbf{h}_1, \mathbf{h}'_1)$, $(\mathbf{h}_2, \mathbf{h}'_2)$, and $(\mathbf{h}'_1, \mathbf{h}'_2)$ fall under Case 1 considered above. The remaining cases are handled similarly to Cases 1 and 2. The conclusion is that, in each case, (B.10) leads to a violation of Condition 2 of Proposition B.1. This means that the inequality (B.10) cannot be true, and so $\psi(\mathbf{h}, S)$ is independent of $\mathbf{h}$, for $\mathbf{h} \in \{0,1\}^N \backslash \{\mathbf{0}, \mathbf{1}\}$. ■

■ B.3 Proof of Lemma 4.3.

Consider the earliest time instant t_1 during the evolution of the SIDE when one of these α-crossings $\{g_1, \dots, g_\nu\}$ disappears. Let us denote the region being removed at t_1 by (i, j). Suppose that, at time t_1^-, there are at least $\nu + 2$ α-crossings remaining. We will presently show that the only α-crossings among $\{g_1, \dots, g_\nu\}$ that can be disappearing at t_1 are either g_1 or g_ν, but not both.

Case 1. *The signal* $\mathbf{u}(t_1^-)$ *has at least* $\nu + 3$ *α-crossings;* $i - 1$ *and* j *are consecutive elements of the set* $\{g_1, \dots, g_\nu\}$.

Since $\mathbf{u}(t_1^-)$ has at least $\nu + 3$ α-crossings whereas the set $\{g_1, \dots, g_\nu\}$ has only ν elements, there must be at least one pair of consecutive α-crossings of $\mathbf{u}(t_1^-)$, say $i' - 1$ and j', such that

(i) $i' - 1, j' \notin \{g_1, \dots, g_\nu\}$, and

(ii) either the α-crossing of $\mathbf{u}(t_1^-)$ immediately to the left of $i' - 1$ or the α-crossing of $\mathbf{u}(t_1^-)$ immediately to the right of j' is an element of $\{g_1, \dots, g_\nu\}$.

Since the region (i, j) disappears at time t_1 while the region (i', j') stays, the energy (4.5) of the region (i, j) must be smaller: $E_{ij} < E_{i'j'}$, or equivalently,

$$\left| \sum_{n=i}^{j} (u_n - \alpha) \right| < \left| \sum_{n=i'}^{j'} (u_n - \alpha) \right|.$$

Therefore, we can increase $\phi(\mathbf{u}^0, \mathbf{h})$ by changing $\mathbf{h}$ as follows: remove the edges $i - 1$ and j and add edges at $i' - 1$ and j'. This contradicts our assumption that $\mathbf{h}$ is the best hypothesis with ν or fewer edges. The conclusion is that two consecutive α-crossings from $\{g_1, \dots, g_\nu\}$ cannot be erased.

Case 2. $i - 1 \notin \{g_1, \dots, g_\nu\}$, *and* $j \in \{g_1, \dots, g_\nu\}$.

Suppose that there are α-crossings of $\mathbf{u}(t_1^-)$ to the left of $i - 1$. Let $i' - 1$ be the α-crossing of $\mathbf{u}(t_1^-)$ immediately to the left of $i - 1$. Since the region (i, j) is removed, we have: $E_{ij} < E_{i',i-1}$, which means that changing $\mathbf{h}$ by moving the edge from j to $i' - 1$ will increase $\phi(\mathbf{u}^0, \mathbf{h})$, and so $\mathbf{h} \neq \mathbf{h}^*_{\leq\nu}(\mathbf{u}^0)$. Thus, it must be that either $i = 1$ or $i - 1$ is the leftmost α-crossing of $\mathbf{u}(t_1^-)$, and therefore $j = g_1$.

Case 3. $i - 1 \in \{g_1, \dots, g_\nu\}$, *and* $j \notin \{g_1, \dots, g_\nu\}$.

This case is handled similarly to Case 2.

Cases 1, 2, and 3 combined imply that, at time t_1, only one of the α-crossings $\{g_1, \dots, g_\nu\}$ can disappear: either g_1 or g_ν. Without loss of generality, suppose that it is g_1. We showed that it can only disappear at the time t_1^- when region (i, g_1) gets erased, where $i = 1$ or $i - 1$ is the leftmost α-crossing of $\mathbf{u}(t_1^-)$. In what follows, we consider only the case $i = 1$; the other case is handled similarly.

We now show that no other α-crossing from the set $\{g_1, \dots, g_\nu\}$ can disappear before t^+. We consider the earliest time instant $t_2 > t_1$ when one of the remaining α-crossings $\{g_2, \dots, g_\nu\}$ disappears. Let us again denote the region being removed by

(i, j). Suppose that, at time t_2^-, there are at least $\nu + 1$ α-crossings remaining.
Case 4. *The indices* $i - 1$ *and* j *are consecutive elements of the set* $\{g_2, \dots, g_\nu\}$.
Case 5. $i - 1 \notin \{g_2, \dots, g_\nu\}$, *and* $j \in \{g_2, \dots, g_\nu\}$.
Case 6. $i - 1 \in \{g_2, \dots, g_\nu\}$, *and* $j \notin \{g_2, \dots, g_\nu\}$.
Cases 4, 5, and 6 are handled similarly to Cases 1, 2, and 3, respectively, with the result that only (i, g_2) or (g_ν, j) can be removed, where $i - 1$ is either 0 or the leftmost α-crossing of $\mathbf{u}(t_2^-)$, and j is either N or the rightmost α-crossing of $\mathbf{u}(t_2^-)$.
We now show how to handle the case when $(1, g_2)$ is removed at t_2; all other cases are handled using similar techniques.

Without loss of generality, we suppose that g_1 is an upward edge of $\mathbf{h}$. Then

$$\sum_{n=1}^{g_1} (u_n - \alpha) < 0, \tag{B.12}$$

as otherwise we would be able to improve $\mathbf{h}$ by removing the edge g_1. Also,

$$\sum_{n=g_1+1}^{g_2} (u_n - \alpha) > 0, \tag{B.13}$$

as otherwise removing the edges g_1 and g_2 would improve $\mathbf{h}$. Since the region $(1, g_1)$ disappeared at time t_1 while the α-crossing at g_2 stayed, we have

$$\begin{aligned} E_{1,g_1} &< E_{g_1+1,g_2}, \\ \text{i.e.,} \quad \left| \sum_{n=1}^{g_1} (u_n - \alpha) \right| &< \frac{1}{2} \left| \sum_{n=g_1+1}^{g_2} (u_n - \alpha) \right|. \end{aligned} \tag{B.14}$$

We combine (B.12), (B.13), and (B.14) to get

$$\begin{aligned} E_{1,g_2} &= \left| \sum_{n=1}^{g_2} (u_n - \alpha) \right| \\ &= \left| \sum_{n=g_1+1}^{g_2} (u_n - \alpha) \right| - \left| \sum_{n=1}^{g_1} (u_n - \alpha) \right| \\ &> \frac{1}{2} \left| \sum_{n=g_1+1}^{g_2} (u_n - \alpha) \right| \\ &= E_{g_1+1,g_2}. \end{aligned} \tag{B.15}$$

Choose the region (i', j') of $\mathbf{u}(t_2^-)$ analogously to Case 1. Then, since $(1, g_2)$ gets removed at t_2 and (i', j') does not,

$$E_{i'j'} > E_{1,g_2}.$$

Together with (B.15), this gives

$$E_{i'j'} > E_{g_1+1,g_2}.$$

Thus, $\mathbf{h}$ can be improved by replacing the edges g_1 and g_2 with i' and j', which is a contradiction. ■

■ B.4 Completion of the Proof of Proposition 4.6.

We need to show that the following situation is impossible:

- $\mathbf{h}_\psi = \mathbf{h}^*_{\leq\bar{\nu}}(\mathbf{u}^0) \neq \mathbf{h}^*_{\leq\bar{\nu}-1}(\mathbf{u}^0)$, and
- $\rho_{i^*,j^*} = 2$, i.e., the SIDE's solution goes from $\bar{\nu}+1$ zero-crossings directly to $\bar{\nu}-1$.

We denote the edges of $\mathbf{h}_1 = \mathbf{h}^*_{\leq\bar{\nu}+1}(\mathbf{u}^0)$ by $g_1, \dots, g_{\bar{\nu}+1}$, and show that each of the three possibilities allowed by Proposition 4.4 in the situation above, leads to a contradiction.
Case (ii). $\mathbf{h}^*_{\leq\bar{\nu}}(\mathbf{u}^0)$ *has edges at the locations* $g_2, \dots, g_{\bar{\nu}+1}$.
Then

$$\begin{aligned} \eta(\mathbf{h}^*_{\leq\bar{\nu}}(\mathbf{u}^0)) - \eta(\mathbf{h}_1) &= E_{1,g_1} - \lambda, \\ \eta(\mathbf{h}_2) - \eta(\mathbf{h}^*_{\leq\bar{\nu}}(\mathbf{u}^0)) &= 2E^* - E_{1,g_1} - \lambda, \end{aligned}$$

and so, for $\mathbf{h}^*_{\leq\bar{\nu}}(\mathbf{u}^0)$ to be better (with respect to η) than both $\mathbf{h}_1$ and $\mathbf{h}_2$, we need to have:

$$\begin{aligned} E_{1,g_1} &< \lambda, \text{ and} \\ 2E^* - E_{1,g_1} &> \lambda, \end{aligned}$$

from which it follows that

$$\begin{aligned} 2E^* - E_{1,g_1} &> E_{1,g_1} \Rightarrow \\ E^* &> E_{1,g_1}. \end{aligned}$$

The latter inequality contradicts the definition of E^* as the smallest energy of any region of $\mathbf{u}(t)$.
Case (iii). $\mathbf{h}^*_{\leq\bar{\nu}}(\mathbf{u}^0)$ *has edges at the locations* $g_1, \dots, g_{\bar{\nu}}$.
This case is handled similarly to Case 2.
Case (iv). $\mathbf{h}^*_{\leq\bar{\nu}}(\mathbf{u}^0)$ *has edges at the locations* $\{g_1, \dots, g_{\bar{\nu}+1}\} \backslash \{i^*, j^*\}$, *as well as one edge at some other location* g'_1. *This situation requires considering several sub-cases—see Case 6 of Section B.3. As they are similar, we only treat the one where the region* $(1, g'_1)$ *was removed before time* t.
Then

$$\begin{aligned} \eta(\mathbf{h}^*_{\leq\bar{\nu}}(\mathbf{u}^0)) - \eta(\mathbf{h}_1) &= 2E^* - E_{1,g'_1} - \lambda, \\ \eta(\mathbf{h}_2) - \eta(\mathbf{h}^*_{\leq\bar{\nu}}(\mathbf{u}^0)) &= E_{1,g'_1} - \lambda. \end{aligned}$$

If $\mathbf{h}_{\psi} = \mathbf{h}^*_{\leq\bar{\nu}}(\mathbf{u}^0)$, we then must have

$$\begin{aligned} E_{1,g_1'} &> \lambda, \text{ and} \\ 2E^* - E_{1,g_1'} &< \lambda, \end{aligned}$$

from which it follows that

$$\begin{aligned} 2E^* - E_{1,g_1'} &< E_{1,g_1'} \Rightarrow \\ E^* &< E_{1,g_1'}. \end{aligned} \tag{B.16}$$

But the region $(1, g_1')$ disappeared before (i^*, j^*), and therefore

$$E^* \geq E_{1,g_1'},$$

which contradicts (B.16). This concludes the proof. ■

■ B.5 Equivalence of the SIDE (4.1) to a Linear Programming Problem.

We need to show that a solution to the linear program (4.24-4.27) exists and consists of ones and zeros. There are $2N-1$ variables involved $(h_1, \ldots, h_N, r_1, \ldots, r_{N-1})$, and it is easily seen that there are $2N-1$ linearly independent constraints in (4.25-4.27). Moreover, $h_1 = \ldots = h_N = r_1 = \ldots = r_{N-1} = 0$ satisfies all the constraints. Thus, (4.25-4.27) define a non-empty polyhedron in $I\!R^{2N-1}$ with at least one vertex (see [65], Theorem 2.8), which means that a solution to (4.24-4.27) exists. We assume non-degeneracy—i.e., that the solution is unique. This is not necessary for the proof, but it greatly simplifies notation. Now suppose that, for this solution $\mathbf{h}$, there exist p and q such that $1 \leq p < q \leq N-1$, and

$$h_p \neq h_{p+1} = h_{p+2} = \ldots = h_q \neq h_{q+1}. \tag{B.17}$$

We now show that, unless $h_q = 0$ or $h_q = 1$, we can change $\mathbf{h}$ to make (4.24) smaller, and that therefore $\mathbf{h}$ cannot be a solution if $0 < h_q < 1$. We will be changing $h_{p+1}, \ldots, h_q$, and so let us write out the portion of (4.24) which depends on them:

$$-sh_q + \lambda(|h_q - h_p| + |h_q - h_{q+1}|), \tag{B.18}$$

where $s = \sum_{i=p+1}^{q} u_i$.

Case 1. Suppose that

$$h_p < h_q \text{ and } h_{q+1} < h_q. \tag{B.19}$$

Then the h_q-dependent portion of Equation (B.18) is:

$$h_q(2\lambda - s). \tag{B.20}$$

If $2\lambda > s$, make $h_q = \max(h_p, h_{q+1})$, which will reduce h_q and therefore also reduce (B.20). It will also make either $h_p = h_q$ or $h_q = h_{q+1}$, violating our assumption (B.17). If $2\lambda < s$, make $h_q = 1$, which will also reduce (B.20). In the degenerate case $2\lambda = s$, we can go either way without changing the solution. Thus, if (B.17) and (B.19) hold, then $h_q = 1$.

Case 2. $h_p > h_q$ and $h_{q+1} > h_q$. This is handled similarly, with the result that $h_q = 0$.

Case 3. $h_p < h_q$ and $h_{q+1} > h_q$. Then the h_q-dependent portion of Equation (B.18) is:

$$-h_q s. \tag{B.21}$$

If $s > 0$, increase h_q by setting $h_q = h_{q+1}$. If $s < 0$, reduce h_q by setting $h_q = h_p$. Both will reduce (B.21) and violate (B.17). In the degenerate case $s = 0$, we can go either way.

Case 4. $h_p > h_q$ and $h_{q+1} < h_q$. This is handled similarly to Case 3.

The situations when $p = 0$ and/or $q = N$ are also handled similarly. The conclusion is that the optimal solution consists of zeros and ones, which means that the linear program (4.24-4.27) is equivalent to the optimization problem (4.23). ■

Bibliography

[1] L. Alvarez, P.L. Lions, and J.-M. Morel. Image selective smoothing and edge detection by nonlinear diffusion, II. *SIAM J. Numer. Anal.*, 29(3), 1992.

[2] M.S. Atkins and B.T. Mackiewich. Fully automatic segmentation of the brain in MRI. *IEEE Trans. on Medical Imaging*, 17(1), February 1998.

[3] M. Basseville and I.V. Nikiforov. *Detection of Abrupt Changes: Theory and Application.* Prentice Hall, 1993.

[4] D.P. Bertsekas and S.K. Mitter. A descent method for optimization problems with nondifferentiable cost functionals. *SIAM J. Control*, 11(4), November 1973.

[5] M.J. Black, G. Sapiro, D.H. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Trans. on Image Processing*, 7(3), 1998.

[6] C. Bouman and K. Sauer. An edge-preserving method for image reconstruction from integral projections. In *Proc. Conf. on Info. Sci. and Syst.*, pages 383-387, Baltimore, MD, March 1991.

[7] C. Brice and C. Fennema. Scene analysis using regions. *Artificial Intelligence*, 1, 1970.

[8] C.B. Burckhardt. Speckle in ultrasound B-mode scans. *IEEE Trans. on Sonics and Ultrasonics*, SU-25, January 1978.

[9] J. Canny. A computational approach to edge detection. *IEEE Trans. on PAMI*, PAMI-8, 1986.

[10] T.-H. Chang, Y.-C. Lin, and C.-C. J. Kuo. Techniques in texture analysis. *IEEE Trans. on Image Processing*, October 1993.

[11] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms.* MIT Press, 1990.

[12] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proc. ICCV*, pages 694–699, Cambridge, MA, 1995.

[13] R.N. Czerwinski, D.L. Jones, and W.D. O'Brien, Jr. Line and boundary detection in speckle images. *IEEE Trans. on Image Processing*, 7(12), 1998.

[14] G. Dahlquist and A. Bjorck. *Numerical Methods.* Prentice Hall, 1974.

[15] S.Z. Der and R. Chellappa. Probe-based automatic target tecognition in infrared imagery. *IEEE Trans. on Image Processing*, 6(1), 1997.

[16] P. Felzenszwalb and D. Huttenlocher. Image segmentation using local variation. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 98-104, 1998.

[17] A.F. Filippov. *Differential Equations with Discontinuous Righthand Sides.* Kluwer Academic Publishers, 1988.

[18] C.H. Fosgate, H. Krim, W.W. Irving, W.C. Karl, and A.S. Willsky. Multiscale segmentation and anomaly enhancement of SAR imagery. *IEEE Trans. on Image Processing*, 6(1), 1997.

[19] M.G. Fleming, C. Steger, J. Zhang, J. Gao, A.B. Cognetta, I. Pollak, and C.R. Dyer. Techniques for a structural analysis of dermatoscopic imagery. *Computerized Medical Imaging and Graphics*, 22, 1998.

[20] J. Froment and S. Parrino. MegaWave2 software package. CEREMADE, URA CNRS 749 de l'Université Paris-IX Dauphine, 1995.

[21] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Trans. on PAMI*, 14(3), 1992.

[22] J.W. Goodman. Statistical properties of laser speckle patterns. In *Topics in Applied Physics, vol. 9: Laser Speckle and Related Phenomena*, 2nd edition, J.C. Dainty, Editor. Springer-Verlag, 1984.

[23] D.R. Greer, I. Fung, and J.H. Shapiro Maximum-likelihood multiresolution laser radar range imaging. *IEEE Trans. on Image Processing*, 6(1), 1997.

[24] K. Haris, S.N. Efstratiadis, N. Maglaveras, and A.K. Katsaggelos. Hybrid image segmentation using watersheds and fast region merging. *IEEE Trans. on Image Processing*, 7(12), 1998.

[25] B. Hassibi, A.H. Sayed, and T. Kailath. Linear estimation in Krein spaces–part i: Theory. *IEEE Trans. Automatic Control*, 41(1), 1996.

[26] B. Hassibi, A.H. Sayed, and T. Kailath. Linear estimation in Krein spaces–part ii: Applications. *IEEE Trans. Automatic Control*, 41(1), 1996.

[27] G.T. Herman. *Image Reconstruction from Projections: The Fundamentals of Computerized Tomography.* Academic Press, 1980.

[28] D.V. Hinkley. Inference about the change-point in a sequence of random variables. *Biometrica*, 57(1), 1970.

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

[30] B.K.P. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17, pages 185-203, 1981.

[31] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. of Comp. Vis.*, 1:321–331, 1988.

[32] I.B. Kerfoot and Y. Bresler. Theoretical analysis of multispectral image segmentation criteria. *IEEE Trans. on Image Processing*, 8(6), 1999.

[33] R.L. Kettig and D.A. Landgrebe. Classification of multispectral image data by extraction and classification of homogeneous objects. *IEEE Trans. on Geoscience Electronics*, GE-14(1), 1976.

[34] S. Kichenassamy. The Perona-Malik paradox. *SIAM J. Applied Math.*, 57, 1997.

[35] J. Koenderink. The structure of images. *Biological Cybernetics*, 50, 1984.

[36] G. Koepfler, C. Lopez, and J.-M. Morel. A multiscale algorithm for image segmentation by variational method. *SIAM J. Numer. Anal.*, 31(1), 1994.

[37] B. Kosko. *Neural Networks for Signal Processing,* pages 37-61. Prentice Hall, 1992.

[38] H. Krim and Y. Bao. A stochastic diffusion approach to signal denoising. In *Proc. ICASSP*, Phoenix, AZ, 1999.

[39] Y. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3, 1989.

[40] S.G. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, 1998.

[41] D. Marr. *Vision.* W.H. Freeman and Co., 1982.

[42] Vision and Modeling Group, MIT Media Lab. *Vision Texture Database.* http://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html

[43] J.-M. Morel and S. Solimini. *Variational Methods in Image Segmentation.* Birkhauser, 1995.

[44] D. Mumford and J. Shah. Boundary detection by minimizing functionals, I. In *Proc. CVPR*, pages 22–26, San Francisco, June 1985.

[45] K.M. Nagpal and P.P. Khargonekar. Filtering and smoothing in an H^∞ setting. *IEEE Trans. Automatic Control*, 36(2), 1991.

[46] S. Osher and L.I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM J. Numer. Anal.*, 27(4), 1990.

[47] T. Pavlidis. Segmentation of pictures and maps through functional approximation. *Computer Graphics and Image Processing*, 1, 1972.

[48] P. Perona. Orientation diffusions. *IEEE Trans. on Image Processing*, 7(3), 1998.

[49] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on PAMI*, 12(7), 1990.

[50] P. Perona, T. Shiota, and J. Malik. Anisotropic diffusion. In [55].

[51] I. Pollak, A. S. Willsky, and H. Krim. A nonlinear diffusion equation as a fast and optimal solver of edge detection problems. In *Proc. ICASSP*, Phoenix, AZ, 1999.

[52] S. Resnick. *Adventures in Stochastic Processes.* Birkhauser, 1992.

[53] J. Rissanen. *Stochastic complexity in statistical inquiry.* World Scientific, 1989.

[54] R.T. Rockafellar. *Convex Analysis.* Princeton Univerity Press, 1970.

[55] B.M. ter Haar Romeny, editor. *Geometry-Driven Diffusion in Computer Vision.* Kluwer Academic Publishers, 1994.

[56] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 1992.

[57] G. Sapiro. From active contours to anisotropic diffusion: connections between basic PDE's in image processing. In *Proc. ICIP*, Lausanne, 1996.

[58] K. Sauer and C. Bouman. Bayesian estimation of transmission tomograms using segmentation based optimization. *IEEE Trans. on Nuclear Science*, 39(4), August 1992.

[59] K. Sauer and C. Bouman. A local update strategy for iterative reconstruction from projections. *IEEE Trans. on Signal Processing*, 41(2), 1993.

[60] P. Schmid. Segmentation of digitized dermatoscopic images by two-dimensional color clustering. *IEEE Trans. on Medical Imaging*, 18(2), February 1999.

[61] U. Shaked and Y. Theodor. H_∞-optimal estimation: A tutorial. In *Proc. of the 31st CDC*, Tucson, Arizona, 1992.

[62] G. Strang. *Linear Algebra and Its Applications,* 3rd Edition. Harcourt Brace Jovanovich, 1988.

[63] P.C. Teo, G. Sapiro, and B. Wandell. Anisotropic smoothing of posterior probabilities. In *Proc. ICIP*, Santa Barbara, CA, 1997.

[64] P.C. Teo, G. Sapiro, and B. Wandell. Creating connected representations of cortical gray matter for functional MRI visualization. *IEEE Trans. on Medical Imaging*, 16(6), December 1997.

[65] J. Tsitsiklis. *Introduction to Mathematical Programming.* 6.251J Course Notes, MIT, 1995.

[66] H. van Trees. *Detection, Estimation, and Modulation Theory*, volume 1. Wiley, 1968.

[67] V.I. Utkin. *Sliding Modes in Control and Optimization.* Springer-Verlag, 1992.

[68] P. Viola and W.M. Wells III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2), 1997.

[69] J. Weickert. Nonlinear diffusion scale-spaces: from the continuous to the discrete setting. In *Proc. ICAOS: Images, Wavelets, and PDEs*, pages 111–118, Paris, 1996.

[70] R.F. Wagner, S.W. Smith, J.M. Sandrik, and H. Lopez. Statistics of speckle in ultrasound B-scans. *IEEE Trans. on Sonics and Ultrasonics*, SU-30, May 1983

[71] M. Wertheimer. Laws of organization in perceptual forms. In *A Sourcebook of Gestalt Psychology*, W.B. Ellis, Editor. Harcourt, Brace, and Company, 1938.

[72] A. Witkin. Scale-space filtering. In *Int. Joint Conf. on AI*, pages 1019–1022, Karlsruhe, 1983.

[73] P. Wolfe. A method of conjugate gradients for minimizing nondifferentiable functions. In *Nondifferentiable Optimization*, M.L. Balanski and P. Wolfe, Editors. North-Holland Publishing, 1975.

[74] G. Wyszecki and W.S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae.* Wiley, 1982.

[75] S.C. Zhu and D. Mumford. Prior learning and Gibbs reaction-diffusion. *IEEE Trans. on PAMI*, 19(11), 1997.

[76] S.C. Zhu and A. Yuille. Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Trans. on PAMI*, 18(9), 1996.

Index