Von Pixeln zu Regionen:
Partielle Differentialgleichungen in der Bildanalyse

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

Thomas Brox

Saarbrücken
2005

Mathematische Bildverarbeitungsgruppe, Fakultät für Mathematik und Informatik
Universität des Saarlandes, 66041 Saarbrücken

Tag des Kolloquiums

26.04.2005

Dekan

Prof. Dr. Jörg Eschmeier

Prüfungsausschuss

Prof. Dr. Hans-Peter Seidel (Ausschussvorsitzender)
Max-Planck-Institut für Informatik, Saarbrücken

Prof. Dr. Joachim Weickert (1. Gutachter)
Universität des Saarlandes

Prof. Dr. Rachid Deriche (2. Gutachter)
INRIA Sophia-Antipolis, Frankreich

Dr. Martin Welk
Universität des Saarlandes

Zusammenfassung

Die vorliegenden Arbeit beschäftigt sich mit Anwendungen partieller Differentialgleichungen in der Bildanalyse. Dabei stehen Anwendungen im Vordergrund, die sich zur Bildsegmentierung verwenden lassen. Dies schließt unter anderem nichtlineare Diffusion, Bewegungsschätzung und die Bildsegmentierung selbst ein. Von Kapitel zu Kapitel werden die verwendeten Methoden dabei mehr und mehr auf die Bildsegmentierung ausgerichtet. Werden in Kapitel 2 noch allgemeine Entrauschungs- und Bildvereinfachungsoperationen vorgestellt, behandelt Kapitel 4 die schon etwas speziellere Aufgabe, Textur und Bewegung aus Bildern zu extrahieren, um entsprechende Merkmale schließlich in Kapitel 5 zur Segmentierung von Bildern verwenden zu können. Dabei zieht sich der Weg von den rohen Bilddaten, den Pixeln, hin zur abstrakteren Beschreibung von Bildern mit Hilfe von Regionen als roter Faden durch die gesamte Arbeit. Dass sich Bildverarbeitungstechniken auch in Forschungsgebieten fern herkömmlicher Bilder als nützlich erweisen können, zeigt Kapitel 3. Hier werden Bildverarbeitungstechniken zur Verbesserung numerischer Verfahren für Erhaltungsgleichungen der Physik verwendet.

Konzeptionell legt diese Arbeit Wert darauf, möglichst viele verschiedene Merkmale zur Segmentierung zu verwenden. Darunter fallen neben den bildgestützten Merkmalen wie Textur und Bewegung auch die wissensbasierte Information eines dreidimensionalen Oberflächenmodells. Die prinzipielle Idee hinter diesem Konzept ist, die Entscheidungsgrundlage zur Trennung von Objektregionen auf eine möglichst breite Informationsbasis zu stellen und somit die Anzahl der Situationen, in denen das Verfahren zufriedenstellende Segmentierungsergebnisse liefert, zu erhöhen.

Ein weiteres Grundkonzept, das in dieser Arbeit verfolgt wird, ist die Verwendung von Coarse-To-Fine-Strategien. Sie kommen sowohl bei der Bewegungsschätzung in Kapitel 4 als auch in der Segmentierung in Kapitel 5 zum Einsatz. In beiden Fällen hat man es mit Optimierungsproblemen zu tun, die viele lokale Optima aufweisen. Herkömmliche lokale Optimierung führt daher meist zu Ergebnissen, deren Qualität stark von der Initialisierung abhängt. Diese Situation lässt sich häufig entschärfen, wenn man das entsprechende Optimierungsproblem zunächst deutlich vereinfacht und erst nach und nach das ursprüngliche Problem zu lösen versucht.

Daneben enthält diese Arbeit viele wesentliche technische Neuerungen. In Kapitel 2 wird nichtlineare Diffusion mit unbeschränkten Diffusivitäten betrachtet, was auch Total-Variation-Flow (TV-Flow) mit einschließt. Eine genaue Analyse von TV-Flow führt dabei zu einer analytischen Lösung, mit Hilfe derer man zeigen kann, dass TV-Flow im diskreten, eindimensionalen Fall exakt identisch mit dem ensprechenden Variationsansatz der TV-Regularisierung ist. Desweiteren werden verschiedene numerische Verfahren in Bezug auf ihre Eignung für Diffusionsfilter mit unbeschränkten Diffusivitäten untersucht.

Man kann TV-Flow als eine Alternative zur Gaußglättung ansehen, mit dem entscheidenden Unterschied, dass TV-Flow kantenerhaltend ist. Durch Ersetzen von Gaußglättung durch TV-Flow lassen sich so diskontinuitätserhaltende Varianten bekannter Operatoren wie etwa des *Strukturtensors* entwickeln.

Auch in Kapitel 3 kommt TV-Flow zum Einsatz, wenn es darum geht, numerische Verfahren zur Approximation hyperbolischer Erhaltungsgleichungen durch Bildverarbeitungsmethoden zu verbessern. TV-Flow fällt dabei die Rolle zu, Oszillationen eines Verfahrens zweiter Ordnung zu beseitigen. In einem alternativen Ansatz werden die Approximationseigenschaften eines Verfahrens erster Ordnung durch einen nichtlinearen Rückwärtsdiffusionsfilter verbessert,

indem die numerische Diffusion, die das Verfahren eigentlich stabilisiert, gezielt wieder entfernt wird. Dabei gelingt es durch eine geeignete Stabilisierung der Rückwärtsdiffusion, die positiven Stabilitätseigenschaften des Originalverfahrens zu erhalten.

Kapitel 4 spaltet sich in zwei Teile auf, wobei der erste Teil von der Extrahierung von Texturmerkmalen handelt, während sich der zweite Teil auf Bewegungsschätzung konzentriert. Bei den Texturmerkmalen besteht dabei das Ziel, einen möglichst niederdimensionalen Merkmalsraum zu kreieren, der dennoch sehr gute Diskriminierungseigenschaften besitzt. Das Grundgerüst dieses Merkmalsraums stellt dabei der in Kapitel 2 vorgestellte, auf TV-Flow basierende Strukturtensor dar. Er beschreibt mit der Orientierung, Stärke und Homogenität der Texturierung bereits sehr wichtige Merkmale einer Textur. Daneben wird ein regionenbasiertes, lokales Skalenmaß entwickelt, das zusätzlich die Größe von Texturelementen als Merkmal einbringt. Diese Texturmerkmale werden später in Kapitel 5 zur Textursegmentierung verwendet.

Zur Bewegungsschätzung werden zwei Verfahren vorgestellt. Das eine basiert auf dem in Kapitel 2 eingeführten Strukturtensor und stellt eine Verbesserung vorhandener lokaler Methoden dar. Das andere Verfahren basiert auf einem globalen Variationsansatz und unterscheidet sich von üblichen Variationsansätzen durch die Verwendung einer Gradientenkonstanzannahme. Diese stattet das Verfahren mit der Fähigkeit aus, auch beim Vorhandensein kleinerer lokaler oder globaler Helligkeitsschwankungen gute Schätzergebnisse zu liefern. Daneben ergibt sich aus der Kombination von nicht-linearisierten Konstanzannahmen und einer Coarse-To-Fine-Strategie ein numerisches Schema, das erstmals eine fundierte Theorie zu den sehr erfolgreichen Warping-Verfahren zur Verfügung stellt. Mit der beschriebenen Technik werden Ergebnisse erzielt, die grundsätzlich präziser sind als alles was bisher in der Literatur vorgestellt wurde.

Bei der eigentlichen Bildsegmentierung in Kapitel 5 geht es schließlich, wie bereits erwähnt, hauptsächlich um die Einbringung der in Kapitel 4 entwickelten zusätzlichen Merkmale und um die Verwendung einer Coarse-To-Fine-Strategie. Dies geschieht im Rahmen von regionenbasierten, impliziten Aktiv-Kontur-Modellen, die auf dem Konzept der Level-Sets aufbauen. Dabei werden die Regionenmodelle um nichtparametrische und lokale Beschreibungen der Regionenstatistik erweitert.

Eine weitere Neuerung ist die Erweiterung des Level-Set-Konzepts auf mehrere Regionen. In einem teils hierarchischen Ansatz wird dabei auch die optimale Anzahl der Regionen geschätzt, was eine erhebliche Erweiterung im Vergleich zu herkömmlichen Aktiv-Kontur-Modellen darstellt.

Außerdem wird die Idee vorgestellt, dreidimensionales Objektwissen in der Segmentierung zu verwenden, indem anhand der Segmentierung die Lage des Objekts geschätzt wird und umgekehrt wiederum das projizierte Objektmodell die Segmentierung unterstützt. Die Umsetzung dieser Idee, wie sie in dieser Arbeit beschrieben wird, steht dabei erst am Anfang. Für die Zukunft ergeben sich hieraus noch viele interessanter Aspekte, die es zu untersuchen gilt.

From Pixels to Regions: Partial Differential Equations in Image Analysis

Thesis for obtaining the degree of a doctor of the engineering sciences of the natural-technical faculties of the Saarland University

Thomas Brox

Saarbrücken, Germany

April 27, 2005

Thesis Advisor: Joachim Weickert

Referees: Joachim Weickert, Rachid Deriche, and Thomas Sonar

Mathematical Image Analysis Group, Department of Mathematics and Computer Science
Saarland University, 66041 Saarbrücken, Germany

`www.mia.uni-saarland.de`

The research was supported by the Deutsche Forschungsgemeinschaft (DFG) under the projects WE-2602/1-1 and WE-2602/1-2

Abstract

This work deals with applications of partial differential equations in image analysis. The focus is thereby on applications that can be used for image segmentation. This includes, among other topics, nonlinear diffusion, motion analysis, and image segmentation itself. From each chapter to the next, the methods are directed more and more to image segmentation. While Chapter 2 presents general denoising and simplification techniques, Chapter 4 already addresses the somewhat more special task to extract texture and motion from images. This is in order to employ the resulting features to the partitioning of images finally in Chapter 5. Thus, in this work, one can clearly make out the thread from the raw image data, the pixels, to the more abstract descriptions of images by means of regions. The fact that image processing techniques can also be useful in research areas besides conventional images is shown in Chapter 3. They are used here in order to improve numerical methods for conservation laws in physics.

The work conceptually focuses on using as many different features as possible for segmentation. This includes besides image-driven features like texture and motion the knowledge-based information of a three-dimensional object model. The basic idea of this concept is to provide a preferably wide basis of information for separating object regions and thus increasing the number of situations in which the method yields satisfactory segmentation results.
A further basic concept pursued in this thesis is to employ coarse-to-fine strategies. They are used both for motion estimation in Chapter 4 and for segmentation in Chapter 5. In both cases one has to deal with optimization problems that contain many local optima. Conventional local optimization therefore usually leads to results the quality of which heavily depends on the initialization. This situation can often be eased, if the optimization problem is first significantly simplified. One then tries to solve the original problem by continuously increasing the problem complexity.

Apart from this, the work contains several essential technical novelties. In Chapter 2, nonlinear diffusion with unbounded diffusivities is considered. This also includes total variation flow (TV flow). A thorough analysis of TV flow thereby leads to an analytic solution that allows to show that TV flow is in the space-discrete, one-dimensional setting exactly identical to the corresponding variational approach called TV regularization. Moreover, various different numerical methods are investigated in order to determine their suitability for diffusion filters with unbounded diffusivities.
TV flow can be regarded as an alternative to Gaussian smoothing, though there is the significant difference of TV flow being discontinuity preserving. By replacing Gaussian smoothing by TV flow, one can develop new discontinuity preserving versions of well-known operators such as the structure tensor.

TV flow is also employed in Chapter 3 where the goal is to improve numerical schemes for the approximation of hyperbolic conservation laws by means of image processing techniques. The role of TV flow in this scope is to remove oscillations of a second order method. In an alternative approach, the approximation performance of a first order method is improved by a nonlinear inverse diffusion filter. The underlying concept is to remove exactly the amount of numerical diffusion that actually stabilizes the scheme. By means of an appropriate stabilization of the inverse diffusion process it is possible to preserve the positive stability properties of the original method.

Chapter 4 is separated into two parts. The first part deals with the extraction of texture features, whereas the second part focuses on motion estimation. Goal of the texture extraction method is to derive a feature space that is as low-dimensional as possible but still provides very good discrimination properties. The basic framework of this feature space is the structure tensor based on TV flow presented earlier in Chapter 2. It contains the orientation, magnitude, and homogeneity of a texture and therefore provides already very important features for texture discrimination. Additionally, a region based local scale measure is developed that supplements the size of texture elements to the feature space. This feature space is used later in Chapter 5 for texture segmentation.

Two motion estimation methods are introduced in Chapter 4. One of them is based on the structure tensor from Section 2 and improves existing local methods. The other technique is based on a global variational approach. It differs from usual variational approaches by the use of a gradient constancy assumption. This assumption provides the method with the capability to yield good estimation results even in the presence of small local or global variations of illumination. Besides this novelty, the combination of non-linearized constancy assumptions and a coarse-to-fine strategy yields a numerical scheme that provides for the first time a well founded theory for the very successful warping methods. The described technique leads to results that are generally more accurate than all results presented in literature so far.

As already mentioned, goal of the image segmentation approach in Chapter 5 is mainly to integrate the features derived in Chapter 4 and to utilize a coarse-to-fine strategy. This is done in the framework of region based, implicit active contour models which are set up on the concept of level sets. The involved region models are extended by nonparametric as well as local region statistics.

A further novelty is the extension of the level set concept to multiple regions. The optimum number of regions is thereby estimated by a hierarchical approach. This is a considerable extension of conventional active contour models, which are usually restricted to two regions. Moreover, the idea to use three-dimensional object knowledge for segmentation is presented. The proposed method uses the extracted contour for estimating the pose of the object, while in return the projected object model supports the segmentation. The implementation of this idea as described in this thesis is only at an early stage. Plenty of interesting aspects can be derived from this concept that are to be investigated in the future.

To Ilka

Preface

Alles Gescheite ist schon gedacht worden,
man muss nur versuchen, es noch einmal zu denken

Johann Wolfgang von Goethe (1749-1832)

Like thousands of PhD students every year and all over the world, I provide with this thesis a work which is supposed to make novel contributions, although it is obvious that there are not enough possible breakthroughs in this world to provide only a minor part of all these students with brand new topics for their theses. Otherwise, man would have already crossed the universe, would have designed new, artificial living creatures, and would have genetically modified himself in order to be bright enough for not reelecting presidents who have proven incapable to safely eat a pretzel.

According to Goethe, there is not much hope to have any reasonable new thought, since all prudent thoughts had already been thought before. This absolute statement might not be completely true. At least one may doubt if people in the 18th century already thought about topics like computer science or genetics. However, there is also some truth in Goethe's statement, in the sense that a new thought does not appear from nowhere but is based on many ideas that have already been thought before. The history of science confirms this hypothesis: revolutions are very rare in science; progress usually emerges from an evolutionary process in which many small ideas sum up to a new important insight. In this sense my thesis contains significant novel contributions.

Besides the aspect of novelty, I attached some importance to the intelligibility of this work. I experienced by myself that reading a good PhD thesis is a very elegant way to get fast access to a new topic. However, this implies that the thesis answers the potential questions of the reader instead of pressing the author's history of understanding into the text. This is the reason why I did not copy my papers into this thesis but rewrote the text from scratch.

In case you come to the judgement that I did a good job writing this thesis, you should know that there are lots of people besides me who are responsible for this success.

First of all I want to thank my supervisor Joachim Weickert not only for guiding me into research, but also for leaving me plenty of freedom and to teach me many secondary capabilities, in particular how to write things down in a way that gives other people the chance to understand them as well.
I also thank Christoph Schnörr from the University of Mannheim for guiding me into the topic of image analysis. For some time I had not been convinced whether I was studying the right subject at the right place, until I was washed into his lecture and quickly found out that image analysis was the right thing to do.
Many thanks are also addressed to Rachid Deriche who invited me to his lab at the INRIA Sophia-Antipolis. Although I spent there only three months, this time was very instructive. Without this stay, Chapter 5 and large parts of Chapter 4 would certainly be missing.

I further want to thank the German research foundation DFG for financing my research. Main goal of the project called *Zur Konstruktion neuer Differenzenverfahren durch nichtlineare Viskositäten der Bildverarbeitung* has been the investigation of diffusion filters with unbounded diffusivities and their application for improving numerical schemes in hyperbolic conservation laws. The project has been in cooperation with the group of Thomas Sonar from the University of Braunschweig.

For the joint work I want to thank all my co-authors. Large parts of the research presented in this thesis has been in close cooperation with other people. Without these cooperations many successes would certainly not have been possible. Therefore, I want to thank especially my colleagues Andrés Bruhn and Nils Papenberg for our very fruitful collaboration on optic flow estimation. The same way I thank Mikaël Rousson from the INRIA Sophia-Antipolis for the nice teamwork on image segmentation and for sharing all his experience on level set methods with me. Many thanks also go to Michael Breuß, Andrea Bürgel, and Thomas Sonar from the University of Braunschweig. They provided me with a lot of helpful information concerning hyperbolic conservation laws and thus helped me to create Chapter 3. Further on, I thank Bodo Rosenhahn from the University of Auckland for our inspiring cooperation on joint segmentation and pose estimation. Apart from his scientific input, he saved my world view by confirming that there exist scientists who can really finish something two weeks *before* a deadline. Although we have not yet had the chance to work together more closely, I want to thank Daniel Cremers for all the discussions, which have always inspired me.

Science is not everything in life, even for a scientist, and therefore my final thanks are addressed to those people who always supported me after I packed up work. These are in particular my parents and Ilka who ensured that my mind was free of image analysis thoughts as soon as the weekend started.

Saarbrücken, April 26, 2005 Thomas Brox

Contents

1 Introduction

There is something fascinating about science.
One gets such wholesale returns of conjecture
out of such a trifling investment of fact.

Marc Twain (1835-1910)

This thesis deals with the application of partial differential equations to image analysis problems, in particular image segmentation. The purpose of image analysis is to give the raw data of images a symbolic meaning that fits to a certain model of the real world and can be used for decisions or further processing. This means, the aim is to make a machine *understand* what is in an image. This kind of pattern analysis draws a strong relation to artificial intelligence. Although the analysis is restricted to patterns in images, the present world model has a large influence on how images are to be interpreted.

The nice issue with image analysis is that processes working on visual data are very well imaginable and hence ease the development of new ideas. Being a pretty young research area, image analysis has attracted many researchers in the last two decades. Meanwhile research in this field has created its own scientific community and can rely on a large number of independent scientific journals and conferences.

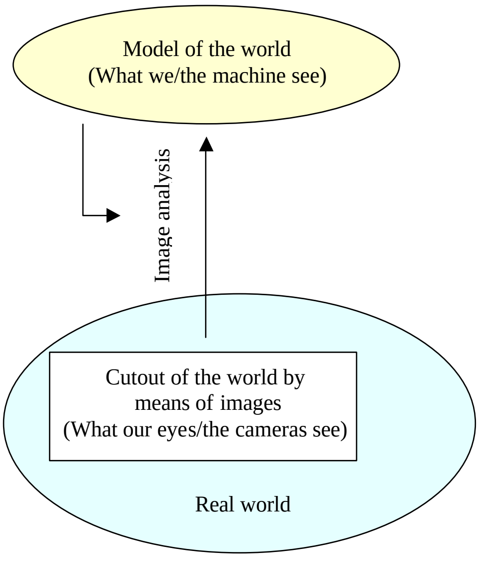


Fig. 1.1: Image analysis uses information from images to update the model of real world.

One motivation that drives many researchers in this field is the goal to finally imitate the performance of the human visual system by a machine. The human's visual capabilities are extraordinary. By blindfolding oneself it becomes vivid how much these capabilities influence our life. With our eyes closed, our orientation is confused, we run against obstacles, cannot read books, screens, and important signs in traffic anymore, cannot tell too much about the mood of people around us, and cannot enjoy the beauties of nature. Vision is an important part of the human creature and causes many problems when it has to be replaced by other senses. Very large parts of the human brain are reserved only for the processing of the information provided by our eyes. This all makes clear that the imitation of these capabilities by a machine is a challenging task and will keep researchers busy probably for the next 20 years, at least. It also explains why progress made in this field often looks trivial for the layman at the first glance, as the same task is accomplished so easily by a human.

After several decades of research, we are just at the beginning of a period where first scientific results find their way to industrial applications and products. The most examples can be found in quality control, automated document processing, and medical applications. In recent time, car industry has further started to work on so-called driver assistance systems which help the driver to keep the car on track by exploiting video data. However, it will still take a few years before such systems will be available in the broad market.
The reason why there are not more computer vision products available is not the lack of possible applications, but the lack of robust methods which work reliably with an acceptable amount of resources. It is simply not possible to install a cluster computer with 5000 processors in a car. On the other hand, it is also not acceptable if plastic bags are confused with children, or vice-versa. However, progress in computer vision research and faster computers extend the number of possible applications year by year.

The main reason why an analysis of the image contents is so difficult lies in the large amount of unrelated data provided by a camera. There are many thousands of pixels in the image supplying their position and their gray value or color, yet they give as such absolutely no hint on the objects in the scene. There is a priori no relation between pixels and objects. The task of image segmentation is exactly to provide such a relationship. This is achieved by partitioning the image into its meaningful parts, i.e., by clustering pixels belonging to possible objects together to regions. Notice that this is a considerable abstraction from the image data and therefore already a big step towards an interpretation of the image. After the extraction of possible objects, it becomes much easier to recognize known objects and to analyze their arrangement in the scene. Thus image segmentation is one of the key issues in image analysis.

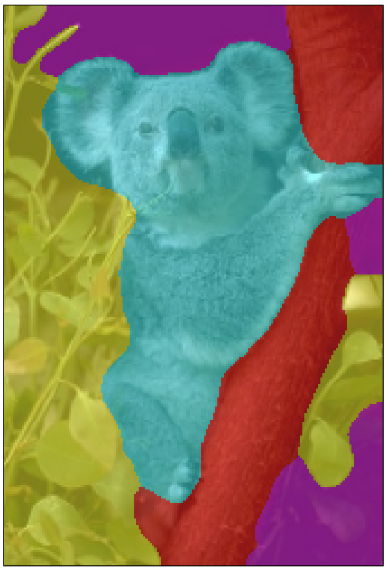


Fig. 1.2: Segmentation example. Pixels are assigned to objects.

The problem with important tasks is that they are often hard to accomplish. Unfortunately, the gray value and the color of pixels are in many cases not sufficient to reliably extract object boundaries. Noise, texture, shadows, highlights, and occlusion, to name only some reasons, greatly disturb image segmentation by means of the gray value or color alone.

There are two strategies to improve the robustness of image segmentation under such circumstances. The first way is to supplement further features apart from the image gray value and color, thereby using the information provided by the image more intensively. This sustains image-driven segmentation, which is often called a bottom-up approach. In the second strategy, the information flow goes the other way round by using a-priori knowledge about what possible objects could look like, therefore preventing some segmentations and enforcing others. This so-called top-down approach is very well suited to cope with disturbances, such as texture, shadows, or highlights, since it can explain them away. It is also the only way to deal with occlusions. On the other hand, the application of top-down knowledge needs an initial guess about possible objects and their position in the image. Otherwise one had to check each entry in an object data base and each pose of these objects. Consequently, the top-down approach only makes sense in combination with a bottom-up strategy.

One important additional feature to be added is texture. Texture is the repeated occurrence of similar local structures, such as the structure of some fabric, a wallpaper, grass, the surface of a lake, or the stripes of a zebra. The supplement of texture features not only allows image segmentation to deal with textured regions. In textured regions the segmentation even becomes more robust with regard to shadows and highlights, since texture is less dependent on illumination.

Texture features can be extracted from the gray value by considering the relationship between neighboring pixels. This means, one is not interested in the actual value of a pixel, but in the difference of this value to those of its neighbors. Integrating this spatial dependency between pixels takes a large amount of additional information into consideration, and thus increases the robustness of segmentation.

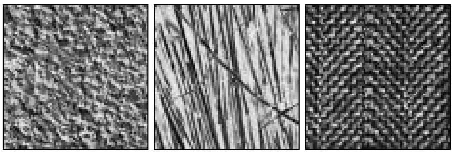


Fig. 1.3: Texture examples.

Another feature that is adequate for extracting object boundaries is motion. Just recall a camouflage animal which is hardly to spot in a scene if we do not know where it is. As soon as it moves, however, we can easily see it. This is because we can use the motion information which is not part of the camouflage concept. Furthermore, the assumption that pixels belonging to the same object move similarly is in most cases satisfied. Thus a significant difference in the motion field can indicate an object boundary.

A natural prerequisite for the computation of motion is the existence of at least two subsequent images. Seeking for correspondences between pixels in the two frames, one extracts the displacement vector which describes the pixel shift from one frame to the next, the so-called *optic flow*. The optic flow vector thus contains the object's motion relative to the camera and projected to the image plane. Although this is not the motion of the object in 3-D space, discontinuities in the displacement field correspond very well to the boundaries of moving objects.

In principle, one could even extract object boundaries in the case of a static object if the camera is moving. This is possible because points in space that are closer to the moving camera yield larger displacements, while points far away from the camera are hardly influenced from the camera's motion. In case of a moving camera, the displacement field therefore contains also a cue for the depth of points in the image. A special case of this scenario is a stereo system. Here one does not seek the displacement between subsequent frames in time, but between the images from two different cameras. This is in fact equivalent to the scenario where a single camera has been moved from one position of the stereo system to the other.

This shows that the supplement of optic flow techniques allows for the utilization of motion as well as depth information and is hence an important issue in image segmentation. Apart from that, optic flow is also important in fields besides image segmentation, in particular for explicitly computing depth and 3-D motion in images.

A further aspect in image segmentation besides the integration of additional information is the representation of regions and their boundaries in the segmentation model. This includes the statistical model for describing the interior of regions.

Further on, a good model for image segmentation is only the first step towards a partitioning of the image. Models for image segmentation, especially if they are based on realistic assumptions, in general lead to rather complex global optimization problems, which need sophisticated techniques in order to find the optimum solution according to the model. From daily life we know: if we are sought to solve a complex problem, it is not sufficient to focus only on some of its local aspects. One first has to get a broader view on the problem to get a hint for a good solution. A similar strategy is also useful in image analysis for solving complex optimization problems like image segmentation. Considering a coarse version of the image often makes the extraction of object boundaries much easier than searching them in the original image. This is because the original data contains lots of distracting details that hinder the focus on the relevant information. These details might not be negligible, yet they often matter only *after* a coarse idea of the final solution has been found.

Outline. Image segmentation, and image analysis in general, are addressed in this thesis by means of PDEs. Chapter 2 provides basic image processing methods. These are in particular the smoothing of data with nonlinear diffusion filters, estimation of orientation in the image, and the framework of variational methods. The descriptions include also numerical issues, such as discretization of continuous formulas to the pixel grid and iterative solvers. Novelties in Chapter 2 contain the theoretical analysis and numerical implementation of a special kind of nonlinear diffusion with unbounded diffusivities, and its application to orientation estimation by means of a nonlinear structure tensor.

Chapter 3 reveals that these techniques are not restricted to image analysis only. In contrast, there is a vivid exchange of techniques with numerical physics going in both directions. One novelty in Chapter 3 is a numerical scheme for approximating hyperbolic conservation laws that is based on an inverse diffusion step motivated from image processing.

After this excursion to a field beyond image analysis, Chapter 4 returns to the image segmentation task by introducing methods for acquiring texture and motion features. A new sparse texture feature space is introduced, which makes use of the nonlinear structure tensor from Chapter 2. Additionally, the texture acquisition method employs a new region based measure for scale.
Chapter 4 further introduces two novel methods for motion estimation. The first method applies again the nonlinear structure tensor from Chapter 2. The second technique is based on a variational approach and introduces new records concerning the accuracy of the estimated motion field. In extensive tests, the limits of this new method are determined.

Chapter 5 finally deals with the image segmentation task itself. The segmentation model is again a variational model, which yields a rather difficult optimization problem. The optimization is performed by means of a continuation method. Thanks to this optimization strategy and the integration of the additional features from Chapter 4, the segmentation technique is capable to extract the contour of the main object in a large set of images. After an extension of the basic segmentation model, it is even possible to extract the contours of multiple objects in the image.
Versus the end of the chapter, the model is further extended by the introduction of shape knowledge. The novelty thereby is the use of *three-dimensional* shape models in 2-D segmentation. This top-down knowledge finally brings image-driven and model-driven information together and builds up a relation between the real three-dimensional world and the two-dimensional cutout of this world provided by the image.

The thesis is concluded by an outlook in Chapter 6. Its purpose is to summarize the work and to show possible improvements of the introduced techniques.

The work further contains three appendices. Appendix A consists of two mathematical proofs emerging from the theoretical analysis of a nonlinear diffusion method called TV flow and its variational counterpart. In the end, the proofs show the equivalence between the diffusion filter and the variational method.
Appendix B lists most of the symbols used in the formulas of this work. This list is intended to be of use especially if the reader does not read a chapter from the beginning and has therefore missed the definition of some symbols in the text. Finally, Appendix C summarizes the contributions of this work and provides a list of all my refereed publications.

2 PDE Based Image Processing

Was nicht translationsinvariant ist,
ist Scheiße in jeder Dimension.

Joachim Weickert
in response to my plan to
use multiplicative splitting schemes

Partial differential equations (PDEs) are one of the most important mathematical tools in image processing and computer vision. As they are appear in diffusion methods as well as in the Euler-Lagrange equations of variational models, they have been considered appear in many image processing tasks. Consequently, this chapter cannot give a complete overview about the field of PDE based image processing. Instead I pick out mainly those methods that will be utilized in the other chapters of this thesis. It will turn out that the methods described in this chapter are not only applicable to image processing or computer vision. In Chapter 3 they will be applied to a field far beyond images, namely in computational physics.

Although the techniques described in this chapter have mostly been introduced already some time ago, new details and modifications are presented. One such detail is the theoretical analysis of *total variation (TV) flow* that has helped, for instance, to develop a local, region based scale measure in Chapter 4 and leads to the equivalence between TV flow and its variational counterpart, TV regularization. Another novelty is the so-called *two-pixel numerics*. Also the edge enhancing diffusivities described in this chapter have not yet been applied in practice. Finally, a nonlinear structure tensor based on these diffusivities, or alternatively TV flow, is proposed. This nonlinear structure tensor will be the neck bone of the texture features described in Chapter 4. So this chapter contains both a review of fundamental techniques that are needed later in this thesis and the presentation of novelties.

2.1 Nonlinear Diffusion

There is a good reason for starting with nonlinear diffusion filtering here right at the beginning: as a powerful denoising and simplification method, it is an important tool in contemporary image processing.

Basically, nonlinear diffusion is a smoothing technique that respects discontinuities in the data – contrary to Gaussian convolution, which could be regarded as the standard approach to smoothing. On the other hand, diffusion is very much related to this standard approach. In fact, diffusion is a generalization of Gaussian smoothing. If discontinuities are neglected in diffusion, one will end up with *homogeneous diffusion*, which comes down to convolutions of the image with a Gaussian kernel.

Diffusion filters are usually modelled in the continuous domain by considering an image as a mapping $I : \mathbb{R}^2 \to \mathbb{R}$. In order to get a physical interpretation, gray values can thereby be regarded as mass concentrations. One then introduces an evolution process with an artificial time variable t. The evolving signal $u(x, y, t)$ is initialized at time $t = 0$ with the image $I(x, y)$ and evolves according to a partial differential equation which describes the type of diffusion process that is applied to the image. For homogeneous diffusion this PDE reads:

$$\partial_t u = \Delta u = \operatorname{div}(\nabla u) \tag{2.1}$$

where Δ denotes the Laplace operator $\partial_x^2 + \partial_y^2$ and ∇ is the gradient operator $(\partial_x, \partial_y)^\top$. Stopping this evolution process after a time T yields a solution, which is equivalent to the result obtained by a convolution of I with the Gaussian kernel K_ρ with standard deviation $\rho = 2\sqrt{T}$ [Iij62, Iij63, WII99]:

$$u(x, y, T) = \left(K_{2\sqrt{T}} * I\right)(x, y) \qquad \text{with } K_\rho(x, y) = \frac{1}{2\pi\rho^2} \exp\left(-\frac{x^2 + y^2}{2\rho^2}\right). \tag{2.2}$$

This formal equivalence of simple Gaussian convolution and diffusion is very helpful, when it comes to applying nonlinear diffusion. Often it is sufficient to simply replace homogeneous diffusion in a method by nonlinear diffusion and thereby taking discontinuities into account. For that purpose, the formalism of the method needs not to be changed and nevertheless one is rewarded by sharp edges.

There are other discontinuity preserving smoothing methods aside from nonlinear diffusion which show a similar behavior. Very popular are for example variational methods [CBFAB97], bilinear filtering [TM98], or statistical methods like M-estimators [CGGM98]. Also wavelet shrinkage can perform the same task as nonlinear diffusion [Don95, DJ94]. Since all these methods are based on the same principle – the preservation of discontinuities – they are very similar in their outcome, and sometimes even direct correspondences can be shown [BSMH98, SW00, CDLL98, WAHM99, SKB01, MWS03, SWB$^+$04, MWB04]. In detail, especially in implementation, however, the methods often differ, and each has its own advantages and drawbacks. One could argue that it is a matter of taste, which method to choose. With its direct relation to Gaussian smoothing, its sound theoretical foundation, and its rotationally invariant continuous model, nonlinear diffusion has many advantages that are important for this work.

Scalar-valued diffusion. Nonlinear diffusion has been introduced in image processing by the famous work of Perona and Malik [PM90]. They considered the scalar-valued case of a rectangular gray value image $I(x, y)$ that serves as initial condition $u(x, y, 0) = I(x, y)$ for the diffusion equation

$$\partial_t u = \operatorname{div}\left(g(|\nabla u|^2)\nabla u\right) \tag{2.3}$$

with homogeneous Neumann boundary conditions, which ensure that no mass (gray value) is lost at the image boundaries. The evolving image u creates a *scale space* [Iij62, Wit83] with scale parameter t, which is called *diffusion time*. Larger diffusion times correspond to more simplified images, i.e. discrete entropy is increasing [SW99].

The *diffusivity* g is a scalar-valued, decreasing function which is responsible for the edge preservation property of nonlinear diffusion. It reduces the amount of smoothing in the presence of edges. Choosing this function is generally the most important degree of freedom

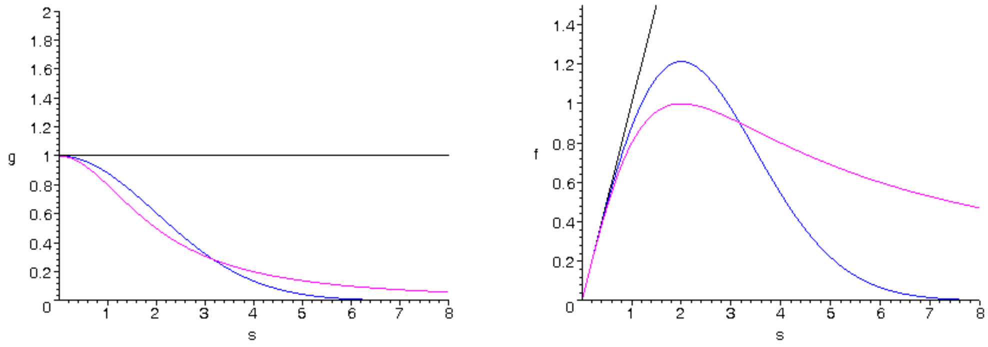


Fig. 2.1: LEFT: Diffusivity functions $g(s^2)$. RIGHT: Corresponding flux functions $f(s) = g(s^2)s$. BLACK: Homogeneous diffusion $g(s^2) = 1$. MAGENTA: Perona-Malik I diffusivity $g(s^2) = \frac{1}{1+s^2/\lambda^2}$ with $\lambda = 2$. BLUE: Perona-Malik II diffusivity $g(s^2) = exp(-\frac{s^2}{2\lambda^2})$ with $\lambda = 2$.

that can change the outcome of nonlinear diffusion considerably. Perona and Malik proposed two different diffusivity functions [PM90]:

$$g(|\nabla u|^2) \quad = \quad \frac{1}{1 + |\nabla u|^2/\lambda^2} \tag{2.4}$$

$$g(|\nabla u|^2) \quad = \quad e^{-\frac{|\nabla u|^2}{2\lambda^2}} \tag{2.5}$$

in which λ is a so-called *contrast parameter*. The role of this parameter becomes clear when restricting (2.3) to the 1-D case and regarding the flux function $f(u_x) = g(|u_x|^2)u_x$. The diffusivity functions as well as the corresponding flux functions for homogeneous diffusion and the variants proposed in [PM90] are depicted in Fig. 2.1. One can see that for homogeneous diffusion the diffusivity is constant, which leads to a mass propagation that is linearly increasing with u_x. For nonlinear diffusion, the diffusivity is reduced in the presence of discontinuities, which leads to a reduced increase of mass transport when u_x becomes larger. From some point on, given by λ, the mass transport even decreases when u_x becomes larger.

Considering the diffusion equation

$$\partial_t u = \partial_x \left(g(|u_x|^2)u_x\right) = f'(u_x)u_{xx} \tag{2.6}$$

one can distinguish cases where $f'(u_x)$ is positive and cases where $f'(u_x)$ is negative. The first case is called *forward diffusion*. With forward diffusion, a pixel always looses more mass to its smaller neighbors than it gains from its larger neighbors, thus edges are always reduced. In the second case, one obtains *backward diffusion*, i.e., a pixel can obtain more mass from its larger neighbors than it looses to its smaller ones. Under such circumstances, edges are enhanced. The contrast parameter λ defines the transition from forward diffusion to backward diffusion, i.e., it steers how large an edge must be for being worth to be preserved.

It has to be mentioned that backward diffusion is ill-posed in the space-continuous setting. This is easy to imagine, since edges can develop to get infinite slope. Fortunately, this is not a problem in practice, as it has been shown that any discretization in space has enough regularizing effects to establish well-posedness [WB97]. Similar conclusions can be found in [Kic97].

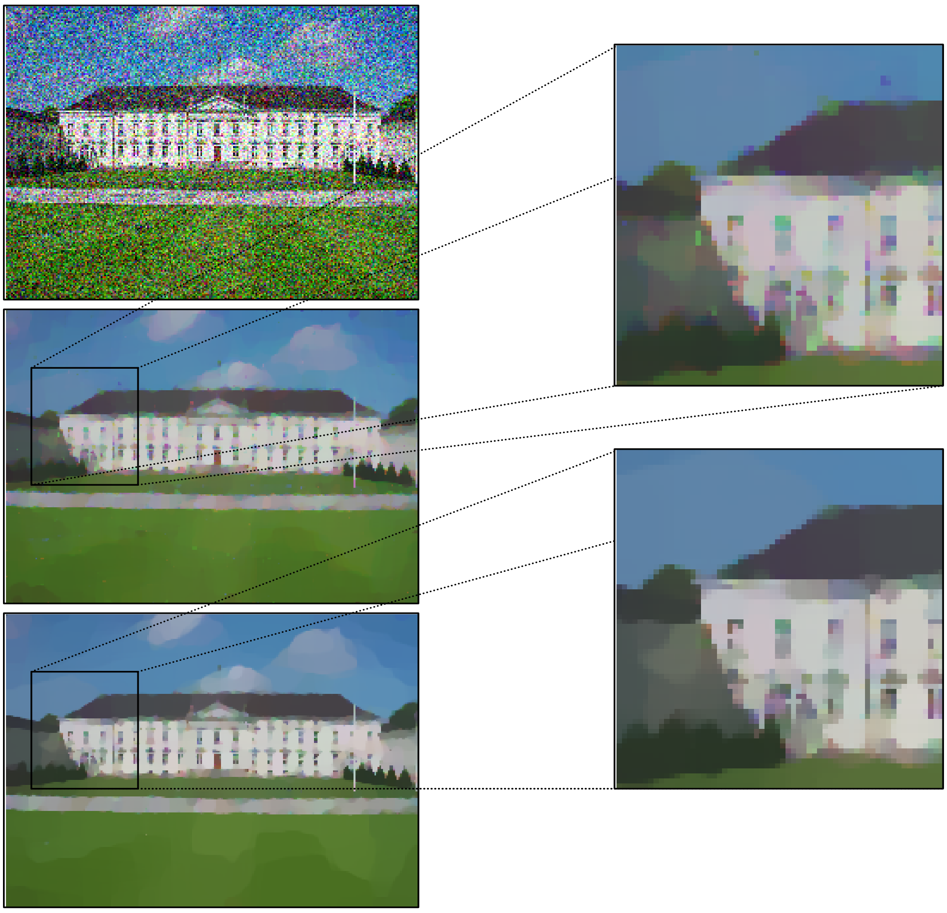


Fig. 2.2: Coupled and non-coupled vector-valued diffusion.
TOP: Input color image with independent Gaussian noise in all three color channels.
CENTER: Scalar-valued diffusion on the three color channels independently.
BOTTOM: Coupled vector-valued diffusion according to (2.7).

Vector-valued diffusion. The diffusion equation (2.3) can only handle scalar-valued input data. Vector-valued data $\mathbf{I} = (I_1, ...I_M)$, as it appears for example in color images, needs some more elaboration. It would be easy to smooth each vector channel separately with (2.3), but this is not the optimum procedure. In such a case edges are preserved – or even enhanced – at different positions in the particular channels. Furthermore, the diffusion in each channel ignores the additional information present in the other channels. In order to fully exploit the available edge information, a coupling between the channels is necessary. In [GKKJ92] such a coupling has been proposed in the form of the diffusion equation

$$\partial_t u_i = \operatorname{div}\left(g\Big(\sum_{k=1}^{M} |\nabla u_k|^2\Big)\nabla u_i\right) \qquad i = 1, ..., M \tag{2.7}$$

with initial condition $\mathbf{u}(x, y, 0) = \mathbf{I}(x, y)$. All vector channels u_i are smoothed here with a joint diffusivity which takes the gradients of all channels into account. Therefore, an edge in one channel also reduces smoothing in the others. Fig. 2.2 shows the better performance of this procedure. A detailed overview on vector-valued smoothing with PDEs can be found in [TD05].

Matrix-valued diffusion. With the scheme for vector-valued data, one could go even a step further and introduce nonlinear diffusion for matrix-valued data. Indeed such applications exist, for instance in medical image processing, where DT-MRI images are to be smoothed [BMB94, Bih95, TD03]. Also the structure tensor, a matrix field that is very frequently used for orientation estimation and that will be explained in detail later in Section 2.2, is a candidate for matrix-valued diffusion.
Again it is reasonable to have a coupling between all matrix channels. In the case of matrix diffusion, the coupling is even more important, since it is necessary to preserve matrix properties, such as positive semi-definiteness [WB02]. A coupled matrix-valued scheme has been introduced in [TD01]:

$$\partial_t U_{ij} = \operatorname{div}\left(g\Big(\sum_{k=1}^{M_1}\sum_{l=1}^{M_2}|\nabla U_{kl}|^2\Big)\nabla U_{ij}\right) \qquad i=1,...,M_1;\ j=1,...,M_2 \tag{2.8}$$

where M_1 is the number of rows and M_2 the number of columns of the evolving matrix field U that has been initialized with the input data. There are further schemes for matrix-valued smoothing, focusing on different goals in smoothing [CTDF04]. An overview of matrix-valued processing can be found in a recent book [WH05].

Anisotropic diffusion. The diffusion schemes discussed so far reduce the amount of smoothing due to edges, so discontinuities are preserved, yet the smoothing is reduced equally in all directions. Sometimes it is better, if only the smoothing across edges is stopped while smoothing along the edge is still performed. This is the idea of anisotropic diffusion [Wei98a], which replaces the scalar-valued diffusivity g by a matrix-valued diffusion tensor D that includes the direction of smoothing. This allows smoothing along the edge with forward diffusion, while at the same time the edge is enhanced by backward diffusion in perpendicular direction. With the same initial conditions as in the respective isotropic case, the diffusion equations for vector and matrix-valued data sets are [Wei98a, WB02]:

$$\begin{aligned}
\partial_t u_i &= \operatorname{div}\left(D\left(\sum_{k=1}^{M}\nabla u_k \nabla u_k^\top\right)\nabla u_i\right) && i=1,...,M && (2.9)\\
\partial_t U_{ij} &= \operatorname{div}\left(D\left(\sum_{k=1}^{M_1}\sum_{l=1}^{M_2}\nabla U_{kl}\nabla U_{kl}^\top\right)\nabla U_{ij}\right) && i=1,...,M_1;\ j=1,...,M_2. && (2.10)
\end{aligned}$$

The diffusion tensor D is computed by applying the diffusivity function g known from isotropic diffusion to the eigenvalues of the matrix $\nabla u \nabla u^\top$. As in the isotropic case it has to be ensured that no flux goes across boundaries.

Anisotropic diffusion is especially useful in cases where the coherence of images is to be improved [Wei99a, Wei99b]. In this scope, the clear distinction between strong smoothing along edges and reduced smoothing across them can be fully exploited. In image simplification tasks this distinction is often not wanted. Furthermore, numerics for anisotropic diffusion are much more challenging. For these reasons, the focus of this thesis will be merely on nonlinear isotropic diffusion.

2.1.1 Total Variation Flow

The choice of the diffusivity function has a large influence on the behavior of nonlinear diffusion, as it defines the extent of communication between neighboring pixels dependent on the difference of their values. In fact, nonlinear diffusion can be more than just smoothing that stops at discontinuities. With an appropriate diffusivity function it can already perform a basic kind of segmentation. One such diffusivity function leads to TV flow [ABCM01, DK00], which has become very popular in recent time:

$$g(|\nabla u|) = \frac{1}{|\nabla u|}. \tag{2.11}$$

TV flow has originally been motivated in the variational framework of *TV regularization* [ROF92, AV94]. Here the idea is to minimize the global total variation $\int_\Omega |\nabla u| dx$ of the result in the image domain Ω while not deviating too much from the original signal. Relations between diffusion and variations denoising are discussed later in Section 2.3. The specific properties of TV flow are:

- Experiments reveal results that are close to piecewise constant segmentations.
- There is no additional contrast parameter in the diffusivity function that has to be optimized (cf. Perona-Malik diffusivities in (2.4) and (2.5)).
- Oscillations are removed with a speed that depends on their spatial scale.
- The signal reaches its average value after a finite diffusion time.
- TV flow works exactly at the limit between forward and backward diffusion.

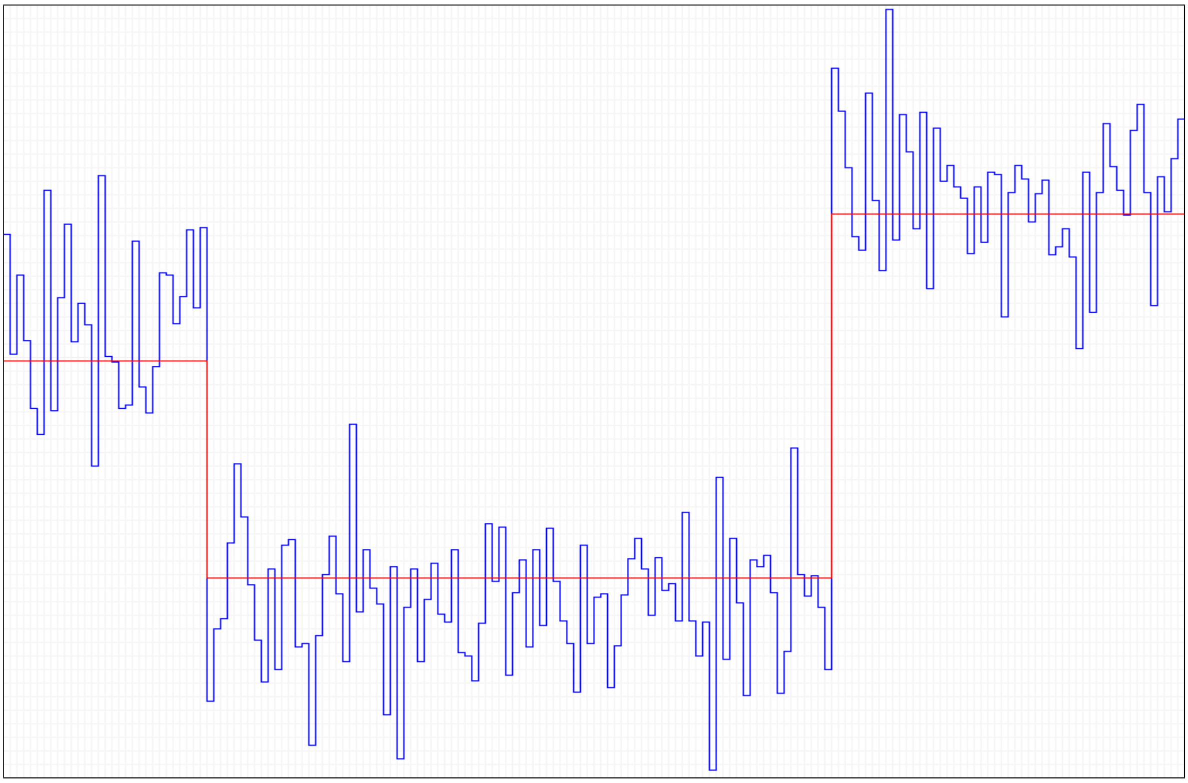


Fig. 2.3: Noise and oscillation removal property of TV flow.
Blue Line: Original noisy signal. Red Line: Signal denoised by TV flow.

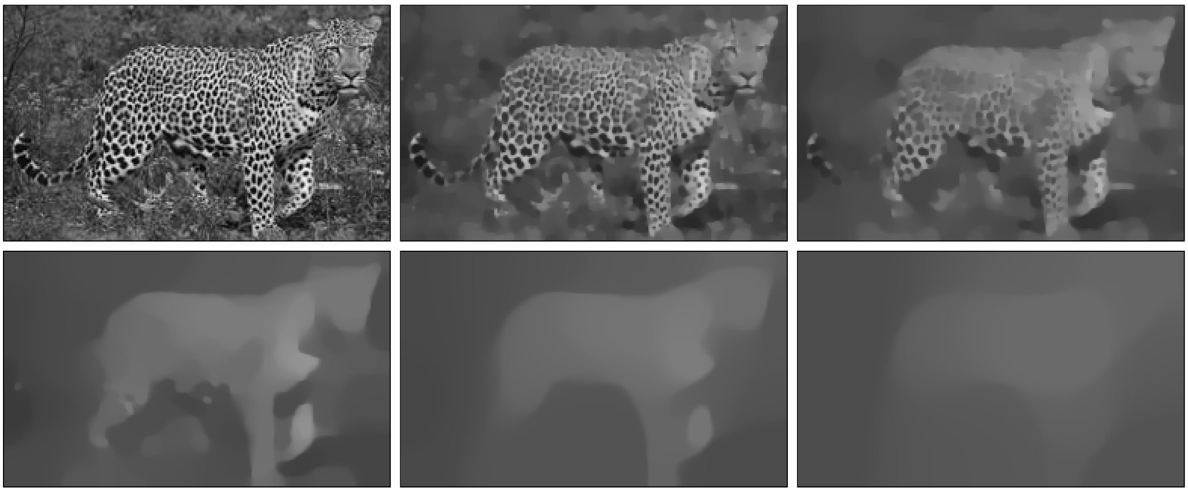


Fig. 2.4: TV scale space.
Upper Row from Left to Right: $t = 0, t = 25, t = 50$.
Lower Row from Left to Right: $t = 100, t = 200, t = 400$.

It is obvious from Fig. 2.3 that noise – here it is Gaussian noise – has mainly an oscillating character that increases the total variation. Thus TV flow, which minimizes the total variation, is very well-suited for removing noise. Furthermore, the segmentation-like behavior, where the number of regions decreases with the diffusion time, is very useful in order to simplify the image data, at least up to a certain degree. This can be observed in Fig. 2.4, which depicts the scale space created by TV flow.

Evolution rules. In the 1-D space-discrete case it is possible to confirm experimental results for TV flow by a theoretical analysis, as it is feasible to compute an analytic solution. It can be proven that TV flow evolves the signal according to the following rules [BWSW03, SWB$^+$04]:

(i) A region of m neighboring pixels with the same value can be considered as one superpixel with mass m. Superpixels never get split.

(ii) The evolution splits into merging events where pixels melt together to larger pixels.

(iii) Extremum pixels adapt their value to that of their neighbors with speed $\frac{2}{m}$.

(iv) The two boundary pixels adapt their value with half that speed.

(v) All other pixels do not change their value.

These rules reveal the very interesting behavior of TV flow. Firstly, in several merging events, it clusters pixels together to regions. So the rules (i) and (ii) endorse the segmentation-like behavior of TV flow. Furthermore, the evolution speed of these regions is inversely proportional to their size (iii). This property ensures that small scale features are removed prior to (meaningful) large scale features and is responsible for the finite extinction time. Finally rule (v) confirms the removal of oscillations, as only extrema in the signal change their value, while non-extrema are not touched at all.
With the evolution rules available, it can further be proven that TV flow is equivalent to its counterpart in the variational framework, i.e. TV regularization [BWSW03, SWB$^+$04],

whereas in general, diffusion and regularization only approximate each other [SW00]. See Section 2.3.4 for more details. Further on, the evolution rules coincide with a spring model for so-called *stabilized inverse diffusion equations (SIDEs)* proposed in [PWK00]. The complete proof of the evolution rules and the equivalence can be found in the appendix of this thesis.

Unfortunately, it has not been possible so far to show similar rules in higher dimensions, especially in 2-D. The 2-D spring model in [PWK00] cannot reproduce TV flow and is not rotationally invariant. In [ACDM02] at least some qualitative properties of TV flow in 2-D could be proven, among them also the finite extinction time. In [BCN02], the behavior of TV flow in some special cases has been investigated in detail.

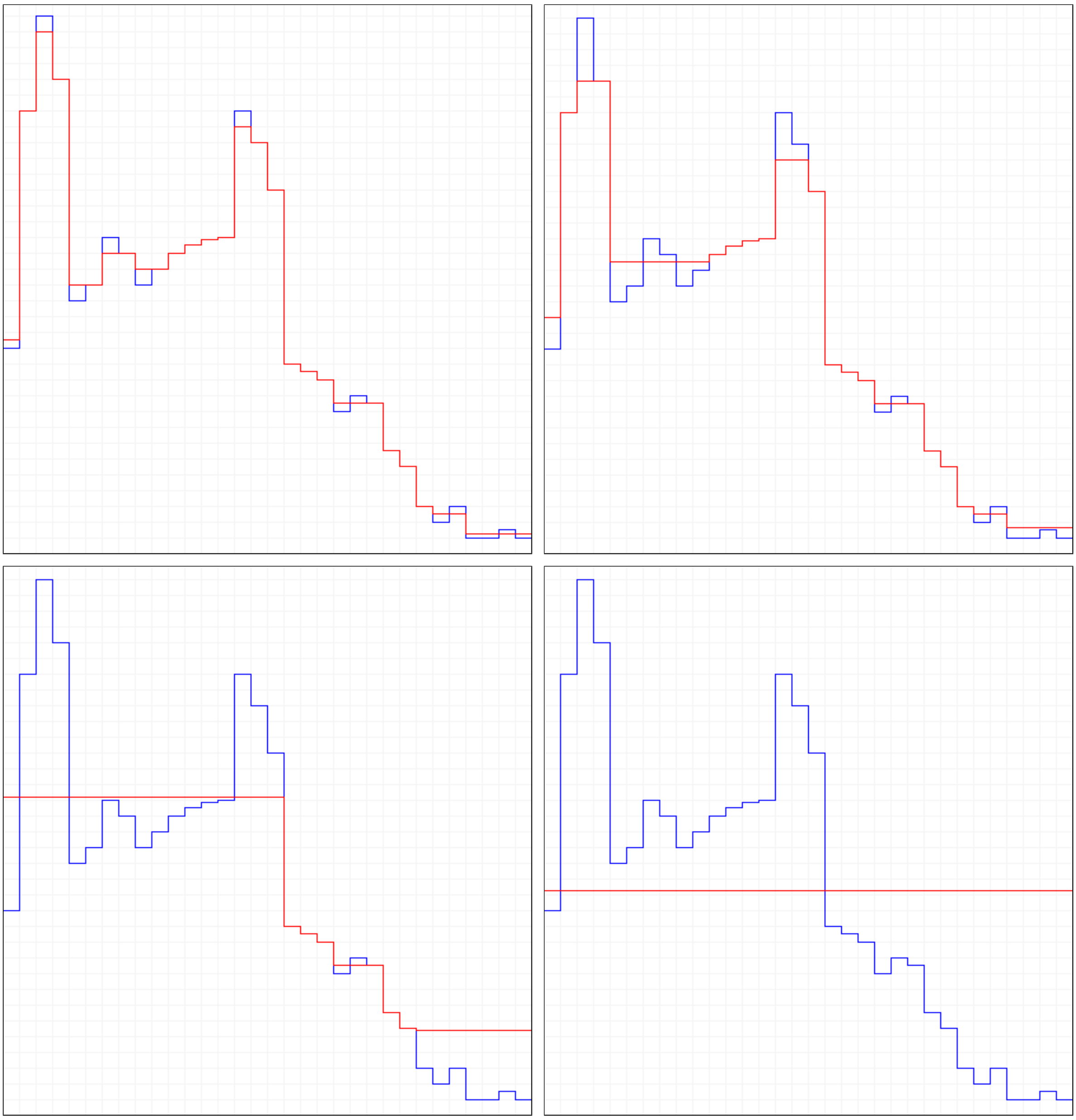


Fig. 2.5: Illustration of the evolution rules of TV flow.
Blue Line: Input signal. Red Line: Evolved signal for $t = 0.5, 2, 25, 130$.

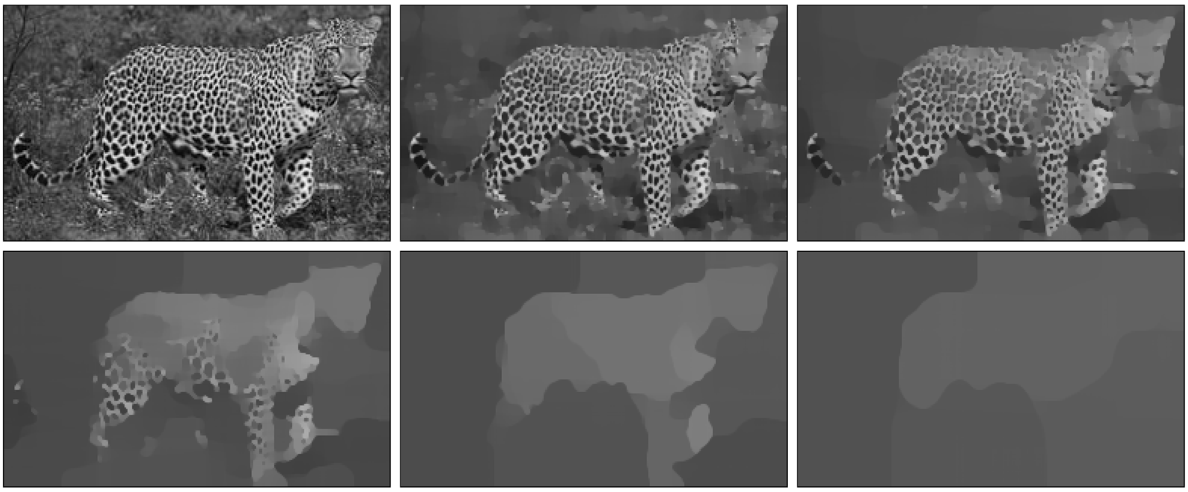


Fig. 2.6: Scale space of edge enhancing flow ($p = 1.5$).
Upper Row from Left to Right: $t = 0, t = 100, t = 200$.
Lower Row from Left to Right: $t = 400, t = 800, t = 1600$.

2.1.2 Edge Enhancing Flow

Although TV flow has lots of nice properties, for some applications its results are still too smooth. This is because TV flow performs no backward diffusion, so it is not able to enhance edges. Therefore, this section introduces a family of diffusivity functions which is like TV flow free of parameters and has similar properties, but allows diffusion in backward direction:

$$g(|\nabla u|) = \frac{1}{|\nabla u|^p} \qquad p \in \mathbb{R}, p \geq 0. \tag{2.12}$$

This class of diffusivities has been considered in [Tsu00], but it has not yet been investigated in the field of image processing or computer vision in practice. It contains homogeneous diffusion and TV flow as special cases for $p = 0$ and $p = 1$, respectively. For $p = 2$ it leads to the so-called *balanced forward-backward diffusion* introduced in [KS02].

The most interesting diffusivities are those for $p > 1$, because they lead to backward diffusion and are therefore edge enhancing. In the sequel, diffusion with such diffusivities will be called *edge enhancing flow.* Whereas for TV flow piecewise constant results can only be guaranteed in 1-D, experiments support the conjecture that the edge enhancing property for $p > 1$ ensures piecewise constant results even for higher dimensions. In Fig. 2.6, one can see the scale space created by edge enhancing flow and $p = 1.5$. In comparison to TV flow, edges become much sharper here. This is because pixels along edges can immediately join a neighboring extremum region due to backward diffusion, thereby sharpening the edge, while with TV flow their value remains unchanged until the evolving extremum region captures them[1]. Consequently, edge enhancing flow keeps the original contrast much better than TV flow, as less smoothing is necessary to obtain a small number of piecewise constant regions. On the other hand, the strongly reduced flux across high gradients disturbs the removal of noise pixels with high contrast. Moreover, backward diffusion often leads to a so-called *stair-casing effect.* This means the diffusion process creates several small regions with enhanced edges out of a ramp function which is actually not wanted. Both effects increase with p.

[1] Recall that in the case of TV flow only extremum pixels change their value. This is because an edge pixel always looses the same amount of mass to its smaller neighbor as it gains from its larger neighbor.

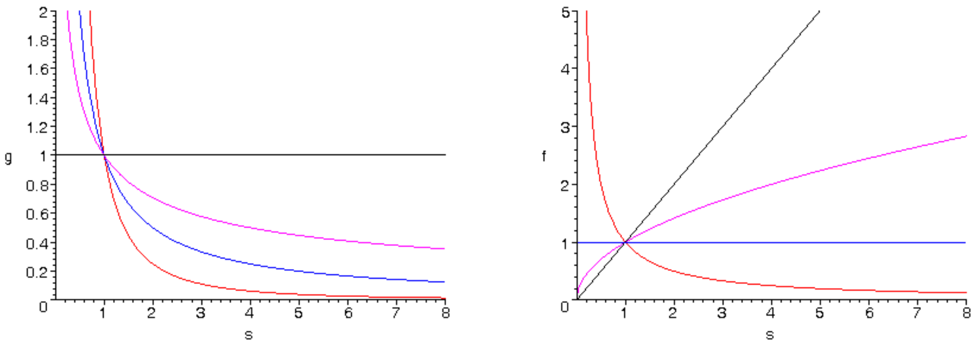


Fig. 2.7: LEFT: Diffusivity functions $g(s^2)$. RIGHT: Corresponding flux functions $g(s^2)s$. BLACK: Homogeneous diffusion $g(s^2) = 1$. BLUE: TV flow $g(s^2) = \frac{1}{s}$. MAGENTA: $g(s^2) = \frac{1}{s^p}$ with $p = \frac{1}{2}$. RED: Edge enhancing flow $g(s^2) = \frac{1}{s^p}$ with $p = 2$.

Fig. 2.7 shows the impact of the parameter p on the diffusivity function as well as the flux function. For $p < 1$ the flux function is monotonously increasing, while for $p > 1$ it is monotonously decreasing. TV flow is the limiting case with constant flux. One can also see that the contrast parameter as present in the Perona-Malik diffusivities is avoided by means of unbounded diffusivities. This way, the diffusion process is either based purely on forward diffusion ($p < 1$) or purely on backward diffusion ($p > 1$). There is no transition between forward and backward diffusion depending on the gradient.

Fig. 2.8 compares the denoising results for different p. As expected, those p that correspond to forward diffusion yield results that have more or less blurred edges. An extreme example is Gaussian smoothing for $p = 0$. On the other hand, those diffusivities with $p > 1$ correspond to backward diffusion, so instead of edge blurring, edges are enhanced. However, one can see that it is also not wise to choose p too large. Apart from the reduced smoothness along edges, diffusion with large p needs large diffusion times to achieve any considerable smoothing at all. Therefore, it is difficult from the numerical point of view to implement these types of diffusion with both high quality and high efficiency.

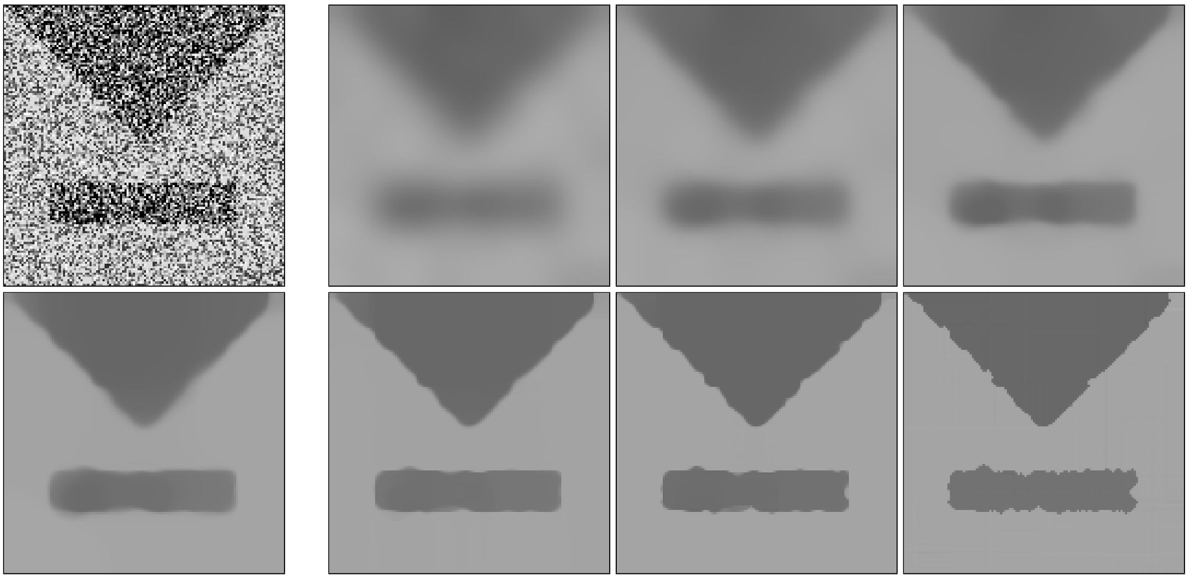


Fig. 2.8: Diffusion results for different p.
LEFT COLUMN: Input image with noise and result obtained with TV flow ($t = 150$).
UPPER ROW: Results with forward diffusion: $p = 0, t = 30$. $p = 0.5, t = 50$. $p = 0.8, t = 80$.
LOWER ROW: Results with backward diffusion: $p = 1.2, t = 300$. $p = 1.5, t = 800$. $p = 2, t = 7000$.

2.1.3 Numerics

Numerics can have a decisive effect on the outcome of nonlinear diffusion regarding both quality and computation speed. This section will focus on the different aspects of the numerical implementation of nonlinear diffusion, with TV flow and edge enhancing flow in particular.

Spatial discretization with central differences. Since the input comes in form of digital images, first a spatial discretization for the continuous partial differential equations has to be found. A typical discretization for nonlinear diffusion is based on spatial derivatives computed by central differences:

$$|\nabla u_{i,j}| \approx \sqrt{\left(\frac{u_{i+1,j}-u_{i-1,j}}{2h_x}\right)^2 + \left(\frac{u_{i,j+1}-u_{i,j-1}}{2h_y}\right)^2}\,. \tag{2.13}$$

As in image processing one usually deals with fixed grids, the spatial grid sizes h_x and h_y are in general set to 1. For the remainder of this thesis it is assumed that $h_x - h_y = 1$.

With this gradient magnitude approximation at a pixel (i, j) it is possible to express the diffusivity $g_{i,j}$. This turns the continuous equation into a space-discrete equation (cf. [Wei98a]):

$$\begin{aligned} \partial_t u &= \partial_x\Big(g(|\nabla u^2|)u_x\Big) + \partial_y\Big(g(|\nabla u^2|)u_y\Big) \\ \frac{du_{i,j}}{dt} &= \frac{g_{i+1,j}+g_{i,j}}{2}\cdot(u_{i+1,j}-u_{i,j}) - \frac{g_{i-1,j}+g_{i,j}}{2}\cdot(u_{i,j}-u_{i-1,j}) \\ &+ \frac{g_{i,j+1}+g_{i,j}}{2}\cdot(u_{i,j+1}-u_{i,j}) - \frac{g_{i,j-1}+g_{i,j}}{2}\cdot(u_{i,j}-u_{i,j-1})\,. \end{aligned} \tag{2.14}$$

A more accurate spatial discretization. The typical discretization based on central differences has disadvantages when accuracy is concerned. It can be observed from the diffusion equation that the diffusivities are actually needed *between* the pixels. This requires to average the computed diffusivities, which implicates some Gaussian smoothing in the process and thus causes a blurring effect. Even worse, also the central differences used for the gradient approximation are not the most precise possibility to approximate the gradient.

If the derivatives are, however, estimated between pixels, it is possible to replace the three-point central differences by two-point differences, with the nice side effect that the diffusivities are given at the right place between the pixels. In 1-D this immediately leads to a discretization scheme:

$$\begin{aligned} |\nabla u_{i+\frac{1}{2}}| &\approx \sqrt{(u_{i+1}-u_i)^2} \\ |\nabla u_{i-\frac{1}{2}}| &\approx \sqrt{(u_i-u_{i-1})^2} \\ \frac{du_i}{dt} &= g_{i+\frac{1}{2}}\cdot(u_{i+1}-u_i) - g_{i-\frac{1}{2}}\cdot(u_i-u_{i-1})\,. \end{aligned} \tag{2.15}$$

Fig. 2.9 shows the gain in accuracy with this scheme in comparison to the discretization based on three-point central differences. While three-point differences cause blurred edges, known as a typical effect of Gaussian smoothing, the scheme based on two-point differences approximates the real diffusion equation much better.

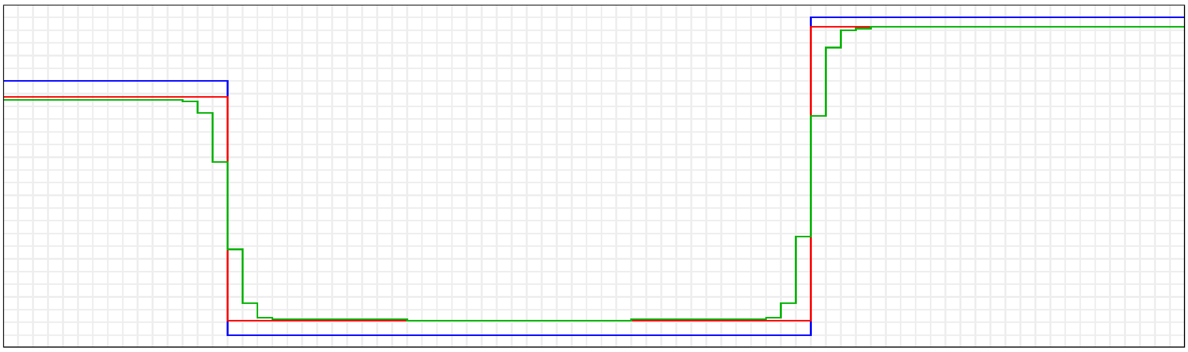
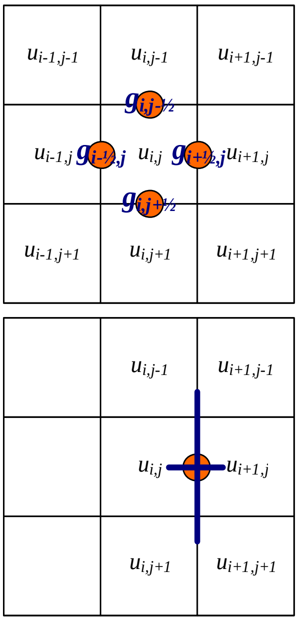


Fig. 2.9: Influence of spatial discretization.
Blue Line: Original signal.
Green Line: Signal evolved with TV flow using central differences.
Red Line: Signal evolved with TV flow using one-sided differences.

In 2-D, a gradient approximation with similar properties needs a bit more elaboration. On a rectangular grid, it is in fact not possible to use merely two-point differences for approximating the diffusivities at the positions where they are needed: half way between the centers of the involved pixels. This is illustrated in the figure on the right. While it is still possible to use two-point differences in one direction, the derivative in the other direction has to be approximated by three-point differences, and even worse, by the average of three-point differences. However, this looks more problematic than it is. The important issue is a high accuracy in direction of the mass flow. The gradient in the other direction is only important to ensure rotation invariance. Its influence on the sharpness of edges is very low. Thus the corresponding discretization in (2.16) yields sharper edges when compared to the discretization based on conventional central differences (see Fig. 2.10).



$$
\begin{aligned}
|\nabla u_{i+\frac{1}{2},j}| &\approx \sqrt{(u_{i+1,j}-u_{i,j})^2+\left(\frac{1}{2}\left(\frac{u_{i+1,j+1}-u_{i+1,j-1}}{2}+\frac{u_{i,j+1}-u_{i,j-1}}{2}\right)\right)^2}\\
|\nabla u_{i-\frac{1}{2},j}| &\approx \sqrt{(u_{i,j}-u_{i-1,j})^2+\left(\frac{1}{2}\left(\frac{u_{i-1,j+1}-u_{i-1,j-1}}{2}+\frac{u_{i,j+1}-u_{i,j-1}}{2}\right)\right)^2}\\
|\nabla u_{i,j+\frac{1}{2}}| &\approx \sqrt{(u_{i,j+1}-u_{i,j})^2+\left(\frac{1}{2}\left(\frac{u_{i+1,j+1}-u_{i-1,j+1}}{2}+\frac{u_{i+1,j}-u_{i-1,j}}{2}\right)\right)^2}\\
|\nabla u_{i,j-\frac{1}{2}}| &\approx \sqrt{(u_{i,j}-u_{i,j-1})^2+\left(\frac{1}{2}\left(\frac{u_{i+1,j-1}-u_{i-1,j-1}}{2}+\frac{u_{i+1,j}-u_{i-1,j}}{2}\right)\right)^2}
\end{aligned}
\tag{2.16}
$$

$$
\frac{du_{i,j}}{dt} = g_{i+\frac{1}{2},j}\cdot(u_{i+1,j}-u_{i,j}) - g_{i-\frac{1}{2},j}\cdot(u_{i,j}-u_{i-1,j}) + g_{i,j+\frac{1}{2}}\cdot(u_{i,j+1}-u_{i,j}) - g_{i,j-\frac{1}{2}}\cdot(u_{i,j}-u_{i,j-1})
$$

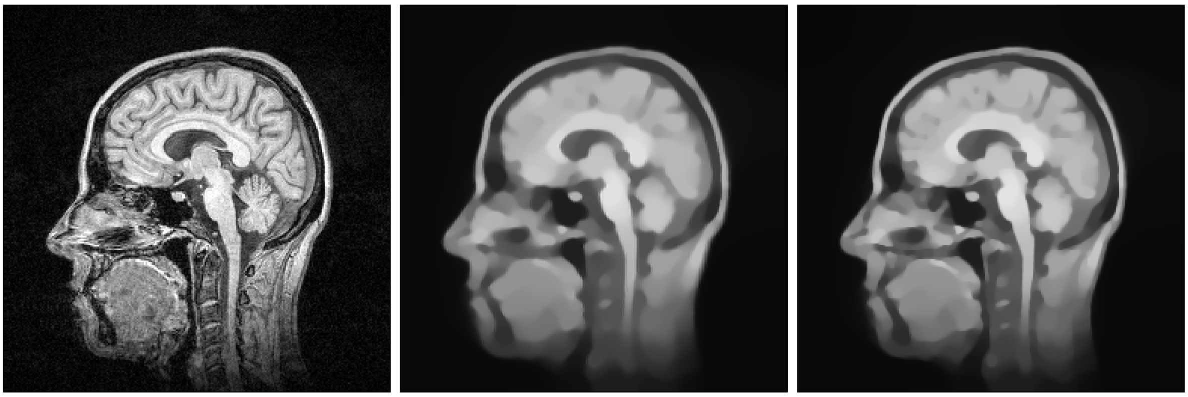


Fig. 2.10: Influence of the spatial discretization in the case of TV flow.
LEFT: Original image. CENTER: Central differences. RIGHT: One-sided differences.
One-sided differences yield sharper edges than standard central differences.

Explicit time discretization. With spatial discretization, the differential equation has been ported to the pixel grid. However, this is not yet a solution of the equation. A numerical solution can be computed by an iterative scheme, which is obtained by discretization in time. The straightforward approach is an explicit scheme with time step size τ:

$$\frac{u^{k+1} - u^k}{\tau} = \operatorname{div}\left(g(|\nabla(u^k)^2|)\nabla u^k\right) \tag{2.17}$$

where u^k denotes u at iteration k. Assumed that g is in between 0 and 1, this scheme is stable for $\tau \leq \frac{1}{2D}$ where D expresses the dimension of the data, e.g. $D = 2$ in case of images [Wei98a]. Although explicit schemes yield accurate solutions, the stability condition makes them rather slow. Further ahead, a more efficient semi-implicit scheme that is stable for arbitrary τ will be discussed.

Stabilization with ϵ-regularization. For stability it has been assumed that g is limited by 0 and 1. TV flow and edge enhancing flow, however, cause unbounded diffusivities when the gradient tends to 0. One can imagine that this happens quite often as the results tend to be piecewise constant. Consequently, the outcome is disturbed by oscillations, as depicted in Fig. 2.11. While for TV flow the oscillations are still bounded, as at least the flux function $f(\nabla u) = g(|\nabla u|^2)\nabla u$ is bounded, for edge enhancing flow the stability problems get really severe, since not only the diffusivity but even the flux can become unbounded and thus the increment in one iteration of (2.17).

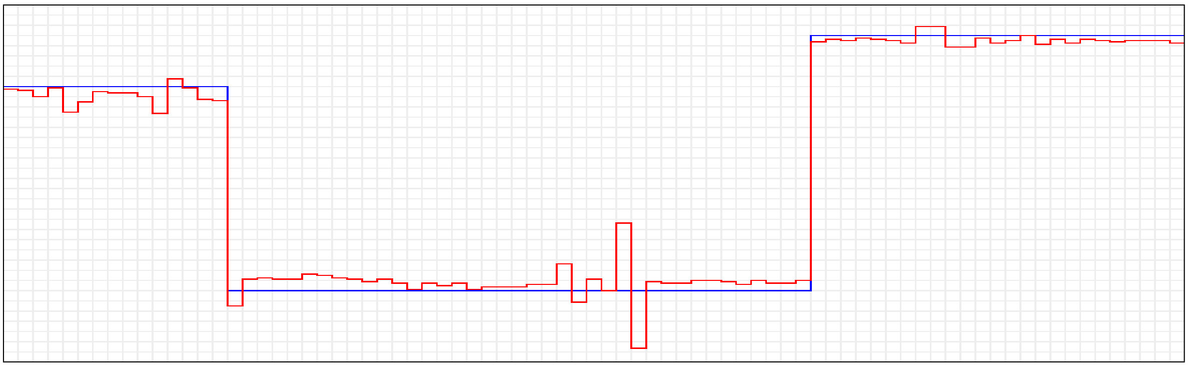


Fig. 2.11: Edge enhancing flow with $p = 1.5$ implemented with an explicit scheme and $\tau = 0.025$.
BLUE: Original signal. RED: Evolved Signal.

A popular solution to this problem developed in the scope of variational TV minimization is the regularization of the diffusivity by a small positive constant ϵ [AV94, COS01, Vog02, FP02], which helps in both the case of TV flow as well as edge enhancing flow:

$$g(|\nabla u|^2) = \frac{1}{(|\nabla u|^2 + \epsilon^2)^{\frac{p}{2}}}. \tag{2.18}$$

The regularization limits the diffusivity at $\frac{1}{\epsilon^p}$ and so stabilizes the explicit scheme of (2.17) with the new stability condition $\tau \leq \frac{\epsilon^p}{2 \cdot D}$. In [FP02] it has been shown for $p = 1$ that in the limiting process, convergence to TV flow can be established.

However, one quickly realizes the drawback: for small ϵ the explicit scheme becomes very slow, as τ has to be chosen correspondingly small causing many iterations to reach a certain diffusion time. Setting ϵ to a larger value, on the other hand, compromises the consistent solution of the original diffusion equation. This ambiguity can only be solved by means of a semi-implicit scheme that is stable for arbitrary τ. Besides that, the ϵ-regularization is a good way to handle unbounded diffusivities. For reasonably small ϵ – values around 0.001 are appropriate for typical images with a data range between 0 and 255 – the regularization does not much harm to the typical behavior of TV flow or edge enhancing flow; see the experiments at the end of this section.

Stabilization with a two-pixel scheme. An alternative way to stabilize diffusion with unbounded diffusivities is based on the mass exchange in the more theoretical case of signals that consist of only two pixels [SW02]. Restricted to only two pixels, it is often feasible to derive an analytic solution of the diffusion equation. Since such an analytic solution does not suffer from numerical instabilities, the key idea of the so-called two-pixel scheme is to make use of two-pixel solutions in order to approximate a stable solution for larger images. The resulting scheme reveals many relations to translation invariant wavelet shrinkage [MWS03, WSM+05].

Consider the family of diffusivities stated in (2.12). With these diffusivities, the diffusion equation in 1-D is:

$$\partial_t u = \left(\frac{u_x}{|u_x|^p} \right)_x .$$

It yields the corresponding discrete evolution equations for two separate pixels u_1 and u_2:

$$\begin{aligned}
\frac{du_1}{dt} &= \frac{u_2 - u_1}{|u_2 - u_1|^p} - \underbrace{\frac{u_1 - u_0}{|u_1 - u_0|^p}}_{=0} = \frac{u_2 - u_1}{|u_2 - u_1|^p} \\
\frac{du_2}{dt} &= \underbrace{\frac{u_3 - u_2}{|u_3 - u_2|^p}}_{=0} - \frac{u_2 - u_1}{|u_2 - u_1|^p} = -\frac{u_2 - u_1}{|u_2 - u_1|^p}.
\end{aligned}$$

The terms containing the dummy pixels u_0 and u_3 are set to 0, since there must be no flux across boundaries. Without loss of generality, suppose that $u_2 > u_1$. Then for the difference $w := u_2 - u_1$ between the pixels it follows:

$$\frac{dw}{dt} = -2w^{1-p}.$$

This differential equation can be solved analytically. It yields

$$\begin{aligned}
w(t) &= w(0)e^{-2t} && \text{for } p = 0 \\
w(t) &= (-2pt + w(0)^p)^{\frac{1}{p}} && \text{for } p > 0.
\end{aligned}$$

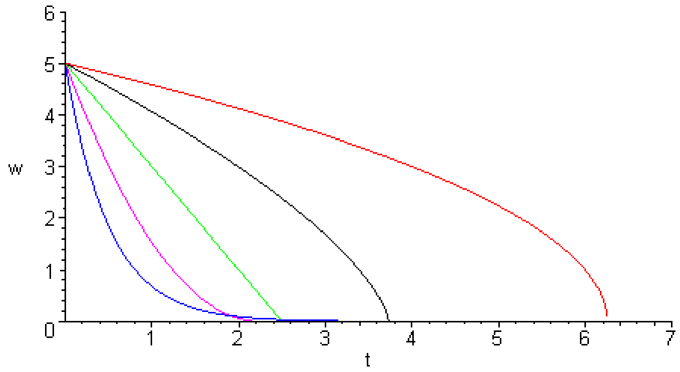


Fig. 2.12: Analytical solution for the difference w between the two pixels u_1 and u_2. BLUE: $p = 0$. MAGENTA: $p = 0.5$. GREEN: $p = 1$. BLACK: $p = 1.5$. RED: $p = 2$.

The solutions for different p are depicted in Fig. 2.12. It can be observed that for all $p > 0$ there is a merging of the pixels after finite time $t = \frac{w(0)^p}{2p}$. At these merging events, the non-stabilized explicit scheme causes overshoots due to a finitely small time step size τ. Now in the case of two pixels, one can intercept the overshoots and correct the values, because for only two pixels it is clear at which value they will meet: namely at their mean value[2]. This yields the following update equation for the pixel u_i in dependence on its partner u_j:

$$u_i^{k+1}(u_j^k) = \begin{cases} u_i^k - \tau \frac{u_i^k - u_j^k}{|u_i^k - u_j^k|^p} & \text{if } \operatorname{sign}(u_i^k - u_j^k) = \operatorname{sign}(u_i^k - 2\tau \frac{u_i^k - u_j^k}{|u_i^k - u_j^k|^p} - u_j^k) \\ \frac{u_i^k + u_j^k}{2} & \text{else.} \end{cases} \tag{2.19}$$

This stabilized two-pixel scheme is so far only a solution for the mass exchange between a pixel and *one* of its neighbors. In order to make the stabilization approach available for larger signals where a pixel has at least two neighbors, it is necessary to integrate the mass exchange resulting from the interaction of a pixel with all its neighbors. A reasonable approach to reduce the operation with multiple neighbors to operations like in (2.19) is by additive splitting [Tai92]:

$$u_i^{k+1} = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} u_i^{k+1}(u_j^k) \tag{2.20}$$

where $\mathcal{N}(i)$ denotes the neighbors of pixel i, and $u_i^{k+1}(u_j^k)$ is the two-pixel solution of pixel i regarding its neighbor j according to (2.19). As one can see from a Taylor expansion, the time step size τ in (2.19) has to be multiplied with the number of neighbors $|\mathcal{N}(i)|$.

The two-pixel scheme is stable for arbitrary time step sizes, as the mass exchange between two pixels is according to the stable analytic solution and averaging is a stable operation. However, it is consistent only for $\tau \to 0$. For large τ, the scheme approximates Gaussian smoothing, as the averaging in (2.20) dominates the evolution. For a detailed consistency analysis the reader is referred to [SWB+04]. Fig. 2.13 shows a comparison between the outcome with large and small time step sizes.

The scheme can be extended to higher dimensions than 1-D, if the approximation of the gradient magnitude $|u_i^k - u_j^k|$ in (2.19) is replaced by the gradient approximation of (2.16). This is necessary in order to ensure rotation invariance. However, one should note that this gradient approximation contains more than two pixels, thus (2.19) is no longer independent from the other pixels. Such an independence can be achieved by considering analytic solutions of four pixels [WWS05].

[2] Diffusion is always mass conservative, i.e. the average gray value is preserved.

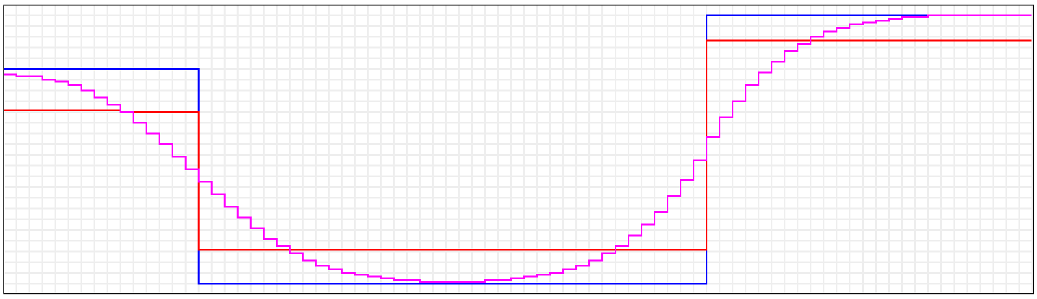


Fig. 2.13: TV flow implemented with the two-pixel scheme and different time step sizes τ. BLUE: Original signal. RED: Evolved signal with $\tau = 0.001$. MAGENTA: Evolved signal with $\tau = 1$.

Semi-implicit time discretization. As already mentioned before, the main drawback of the explicit scheme with ϵ-regularization is the small time step size necessary for stability. Since small time steps come together with many iterations, diffusion implemented via the explicit scheme is computationally very expensive, particularly if unbounded diffusivities come into play and demand even smaller time steps.

The stability condition on the time step size can be lifted with a semi-implicit scheme. For this purpose, consider the discrete diffusion equation in (2.17) of the explicit scheme in matrix-vector notation

$$\frac{u^{k+1} - u^k}{\tau} = A(u^k)u^k \qquad \Rightarrow \qquad u^{k+1} = (I + \tau A(u^k))u^k \tag{2.21}$$

with I being here the $N \times N$ unity matrix, where N is the number of pixels in the image, and $A(u^k)$ being a matrix of the same size with entries:

$$a_{ij} = \begin{cases} g_{i\sim j} & \text{if } \ j \in \mathcal{N}(i) \\ -\sum_{k\in\mathcal{N}(i)} g_{i\sim j} & \text{if } \ j = i \\ 0 & \text{else.} \end{cases} \tag{2.22}$$

Here, $g_{i\sim j}$ denotes the diffusivity between the pixels i and j. The notation reveals that with the explicit scheme the initially nonlinear diffusion equation has been linearized in a time interval of size τ and is solved by performing for each interval a matrix-vector multiplication. A semi-implicit time discretization considers for the multiplication with A the new value u^{k+1} [CLMC92]:

$$\frac{u^{k+1} - u^k}{\tau} = A(u^k)u^{k+1} \tag{2.23}$$

This results in a linear system that has to be solved at each iteration:

$$(I - \tau A(u^k))u^{k+1} = u^k. \tag{2.24}$$

The matrix $(I - \tau A)$ is a sparse, positive definite, and diagonally dominant matrix with one main diagonal and two off-diagonals in the 1-D case. Such a tridiagonal linear system can be solved very efficiently with the *Thomas algorithm* [Tho49]. For its implementation in the scope of nonlinear diffusion see also [WtHV98].

With the employment of the more current data vector u^{k+1}, the semi-implicit scheme is stable for arbitrary time step sizes τ [Wei98a, WtHV98]. However, it should be noted that the linearization in time is still present, so very large time step sizes would compromise the nonlinear character of the diffusion equation, i.e. the diffusivity would only be rarely adapted to the new data. As a remedy, one might be interested in considering the fully implicit time discretization

$$\frac{u^{k+1} - u^k}{\tau} = A(u^{k+1})u^{k+1}. \tag{2.25}$$

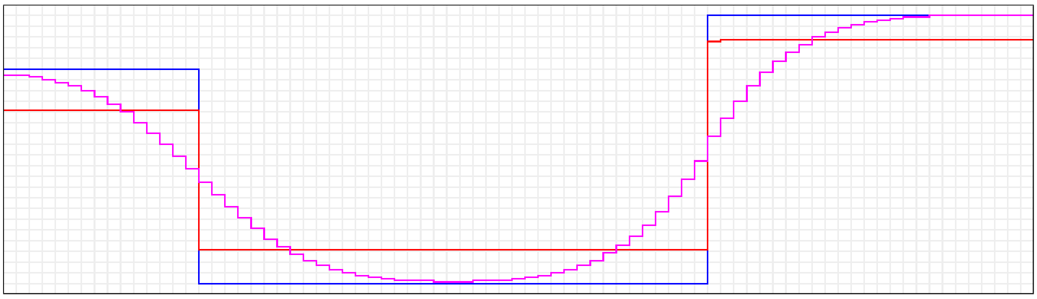


Fig. 2.14: Comparison of methods applicable with large time step sizes τ.
Blue: Original signal. Red: Semi-implicit scheme with $\tau = 1$ and $\epsilon = 0.001$.
Magenta: Two-pixel scheme with $\tau = 1$.

However, this leads to a nonlinear system of equations, which is much more complicated to solve than the linear system stated above. Fortunately, this is also not necessary, as the semi-implicit scheme is already sufficient to guarantee stability. One can also see in Fig. 2.14 that the semi-implicit scheme, combined with ϵ-regularization, yields good solutions even for larger time steps. For such large time steps, it clearly outperforms the two-pixel scheme, which is also stable for arbitrary τ.

The gain in computational efficiency with the semi-implicit scheme in comparison to the explicit scheme is considerable, in particular for small ϵ. One iteration in the semi-implicit scheme takes approximately twice the time of an iteration in the explicit scheme. On the other hand, for e.g. $\epsilon = 0.001$, the time step size can be set at least 2000 times larger while obtaining visually similar results. Thus for a reasonable ϵ, the semi-implicit scheme is, with comparable accuracy, about 1000 times faster than the explicit scheme.

AOS scheme. Unfortunately, the Thomas algorithm cannot be directly employed in higher dimensions than 1-D, since the matrix A gets additional side-diagonals. Therefore, iterative methods for solving linear systems have to be used, like for example *Gauss-Seidel* or *successive over-relaxation (SOR)* [You71].

A faster approach is the so-called *additive operator splitting (AOS)* scheme [Tai92, WtHV98]. Like in the two-pixel scheme, the diffusion of the whole system is split into separate diffusion systems. The result of the whole system is then approximated by the average of the partial solutions (cf. (2.19) and (2.20)). In the scope of the semi-implicit scheme, the diffusion equation of dimension D can be split into one-dimensional diffusion along the coordinate axes, i.e. only neighbors in n-direction are considered in A_n:

$$(I - D\tau A_n(u^k))u_n^{k+1} = u^k \qquad n = 1, ..., D. \tag{2.26}$$

These linear systems can be solved separately with the Thomas algorithm. The solution of the D-dimensional diffusion process is then obtained by

$$u^{k+1} = \frac{1}{D}\sum_{n=1}^{D} u_n^{k+1}. \tag{2.27}$$

Fig. 2.15 shows diffusion results implemented with different time step sizes. Obviously, the splitting does not affect the quality of the results as long as reasonable time step sizes are used. Only if the time steps are chosen too large, one can observe artificial stripes in the result. This is not surprising, as in one iteration mass can only diffuse along the coordinate axes. Thus a certain amount of iterations is necessary to ensure a regular contribution of

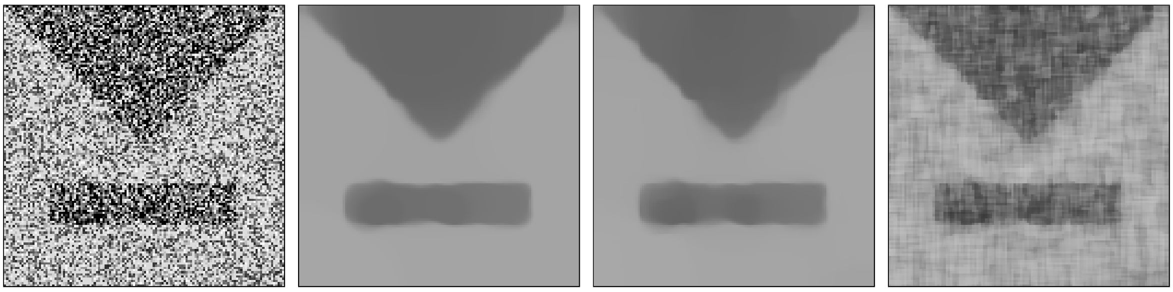


Fig. 2.15: TV flow and $t = 150$ implemented with AOS, $\epsilon = 0.001$, and different time steps τ. From Left to Right: Original image. $\tau = 0.25$. $\tau = 5$. $\tau = 50$.

mass. In the case of backward diffusion, it can even happen that a stripe caused by strong diffusion along a coordinate axis is interpreted as an edge and will be enhanced. Although one should be aware of this effect, it does not really reduce the value of the AOS scheme. It should be noted that AOS has the same approximation order as the semi-implicit scheme without splitting [WtHV98], and unlike multiplicative splitting schemes it treats all coordinate axes equally. A certain amount of iterations, which is also necessary to keep the nonlinear character of the diffusion, is sufficient to avoid any visible artifacts. With such reasonable time steps, e.g. $\tau = 1$ for TV flow, the increase in computational efficiency when compared to the explicit scheme is still one order of magnitude for conventional nonlinear diffusion, and three orders of magnitude for diffusion with unbounded diffusivities and reasonably small ϵ. For further speedups it is possible to implement AOS on a computer cluster. With 256 nodes, for instance, an additional speedup with about factor 200 can be achieved [BJF$^+$04].

Comparison of the numerical schemes. After the introduction of several numerical schemes for TV flow, it is time to finally compare them. This is done by applying TV flow to the signal depicted in Fig. 2.18. For space-discrete, one-dimensional TV flow, there is an exact solution available, so the quality of the numerics can be measured with respect to this ground truth. Therefore, a comparison is only made in 1-D. One can assume that the performance of the numerical schemes approximately carries over to higher dimensions.

Fig. 2.19, Fig. 2.20, and Fig. 2.21 show the diffusion results for the straightforward explicit scheme without stabilization, the same scheme with ϵ-regularization, the two-pixel scheme, and the semi-implicit scheme with ϵ-regularization, each in comparison with the exact solution. Fig. 2.19 reveals the oscillations caused by the scheme without stabilization. The two-pixel scheme and the explicit scheme with ϵ-regularization yield very similar results that tend to be too smooth. The difference between these two schemes becomes better visible when the average L_2 errors toward the exact solution are compared (see Fig. 2.16). Here the two-pixel scheme turns out to yield a higher accuracy for very small time steps. The semi-implicit scheme clearly performs best for large and medium time step sizes. Only if a very high accuracy is wanted, i.e. for very small time steps, the two-pixel scheme is the better choice.

Fig. 2.17 depicts the total variation per pixel. In the exact solution, there is no total variation left after some finite time. This has been mentioned as one of the principal properties of TV flow. It can be observed that the numerical approximations, however, cannot fully reproduce this property. The scheme without stabilization is even after infinite time not able to decrease the total variation below 2τ. This is obviously due to the oscillations, which have in the case of TV flow the magnitude τ caused by the over- and undershoots at merging events.

From the figures it can be concluded that the two-pixel scheme should be preferred, if a very high accuracy is needed and there is no exact solution available. The two-pixel scheme yields the most accurate results for a given computational effort. On the other hand, if the computational efficiency is more important, as it will be the case in most applications, the semi-implicit scheme with one-sided differences and ϵ-regularization is clearly the best choice. Even for quite large time steps, it yields sufficient accuracy to avoid strong visible deviations from the exact solution, and it is some orders of magnitude faster than the other schemes. Thus it will be, in form of AOS in the 2-D case, the numerical scheme used for TV flow and edge enhancing flow throughout the remainder of this thesis.

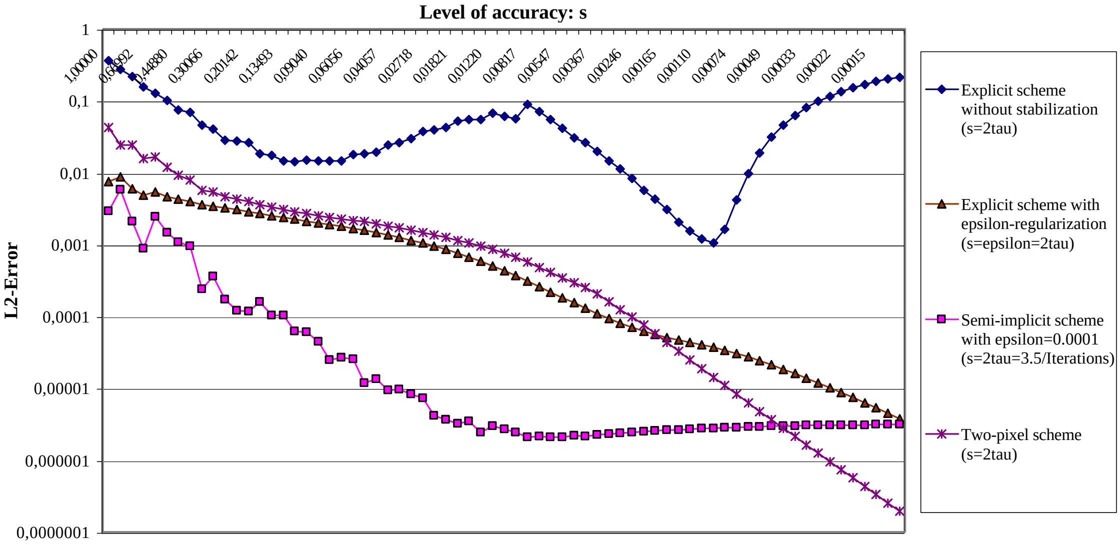


Fig. 2.16: Comparison of the average L_2-error for TV flow and the signal depicted in Fig. 2.18 after diffusion time $t = 1.75$ and for different levels of accuracy.

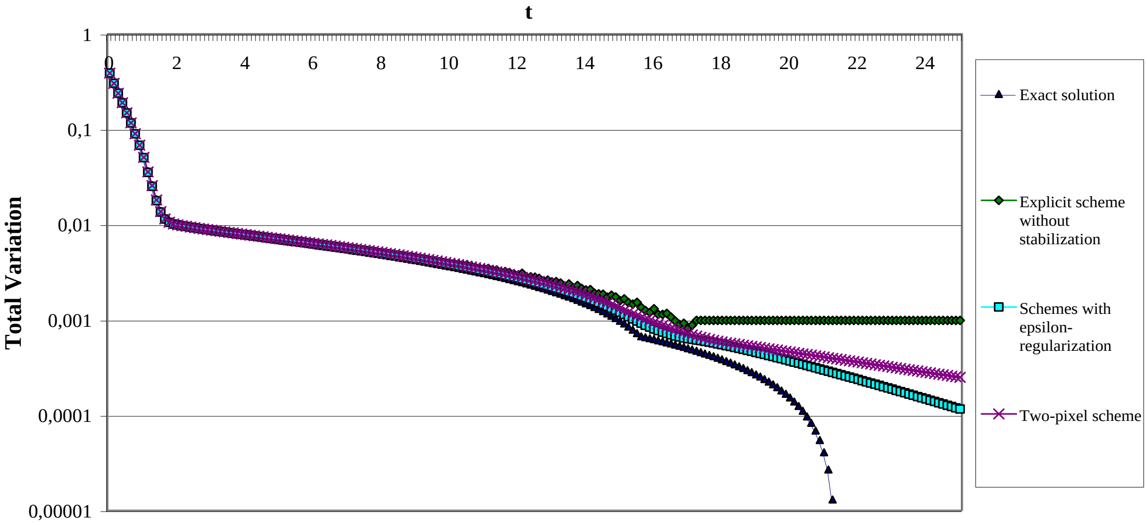


Fig. 2.17: Comparison of the average total variation per pixel for TV flow and the signal depicted in Fig. 2.18 after different diffusion times t. $\epsilon = 2\tau = 0.001$.

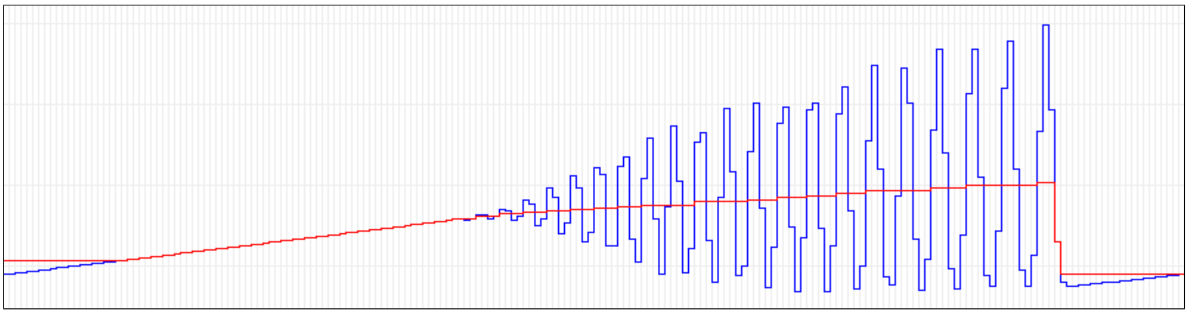


Fig. 2.18: BLUE LINE: Input signal used for comparing the different numerical schemes for TV flow. The values go from 0.67 to 3.97. RED LINE: Exact solution after $t = 1.75$.

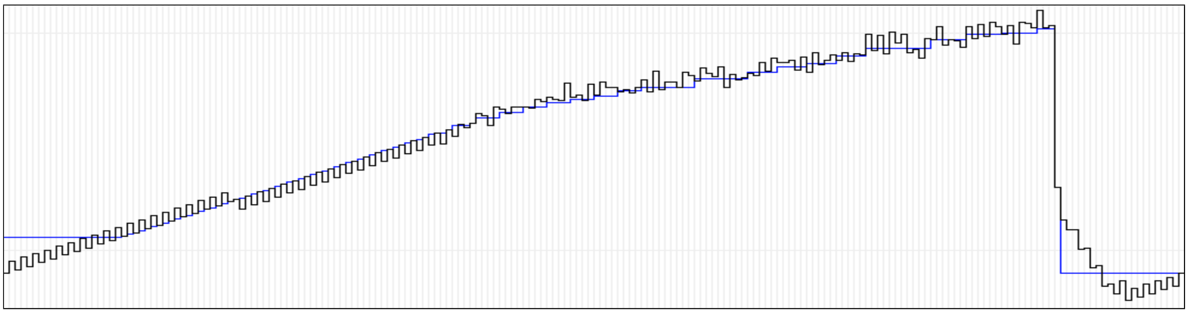


Fig. 2.19: Evolution result for the signal in Fig. 2.18 after diffusion time $t = 1.75$.
BLUE LINE: Exact solution.
BLACK LINE: Explicit scheme without stabilization and $2\tau = 0.05$.

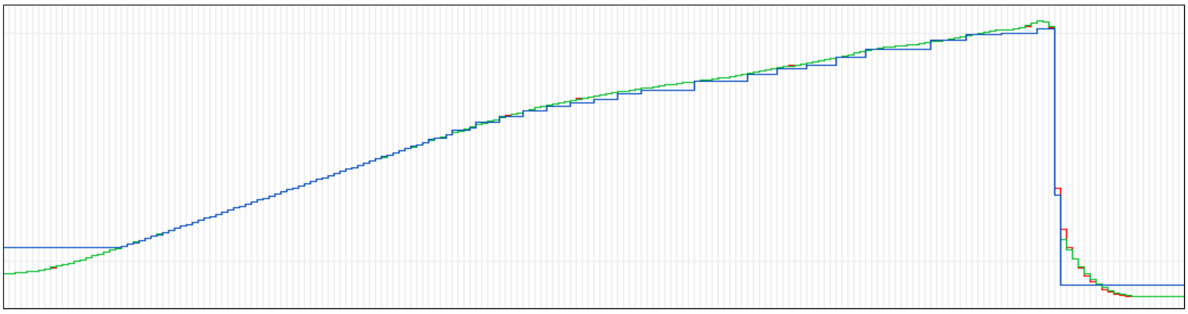


Fig. 2.20: Evolution results for the signal in Fig. 2.18 after diffusion time $t = 1.75$.
BLUE LINE: Exact solution.
RED LINE: Explicit scheme with ϵ-regularization and $\epsilon = 2\tau = 0.05$.
GREEN LINE: Two-pixel scheme with $2\tau = 0.05$.

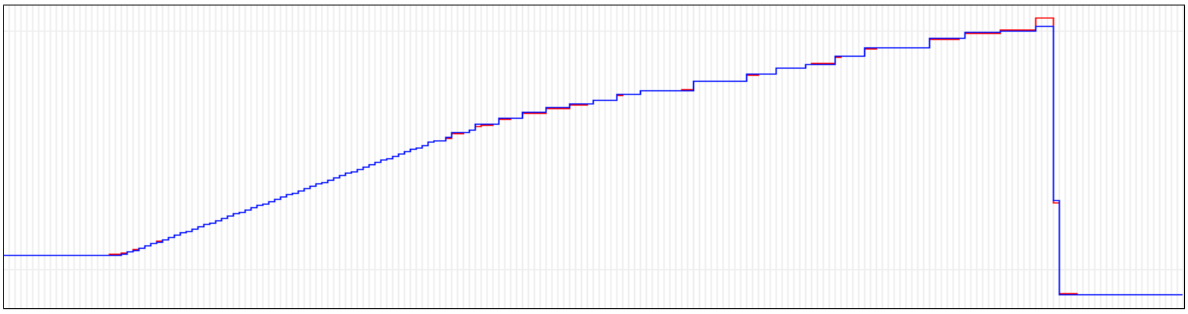


Fig. 2.21: Evolution result for the signal in Fig. 2.18 after diffusion time $t = 1.75$.
BLUE LINE: Exact solution.
RED LINE: Semi-implicit scheme with ϵ-regularization, $\epsilon = 0.001$, and $2\tau = 0.05$.

2.1.4 Summary

In contemporary image analysis, Gaussian convolution is still the dominating smoothing method due to its robustness, its simplicity, its efficient numerics, and the clear meaning of one single parameter. At the end of this chapter about nonlinear diffusion an important statement can be derived: TV flow is as robust under noise as Gaussian smoothing, it can be described by a simple partial differential equation, with the semi-implicit scheme and ϵ-regularization it can be implemented very efficiently and absolutely stable, and finally, like Gaussian smoothing, it has only one single parameter with a clear meaning as scale parameter. So TV flow shows all the nice properties of Gaussian smoothing. Additionally, TV flow respects discontinuities in the data. *Consequently, TV flow should replace Gaussian smoothing whenever discontinuities matter. There is no reason to use Gaussian smoothing in such cases.* Surely, implementation of the numerics for TV flow is much harder than programming of a linear convolution operator. But once implemented, TV flow can be applied in the same simple manner as Gaussian smoothing. The semi-implicit numerics are absolutely stable without any parameter tuning. Setting $\epsilon = 0.001$ and $\tau = 1$ as default values, TV flow can be used without causing any problems, and it yields sharper results than those obtained with Gaussian convolution. There is one reason why Gaussian smoothing remains nevertheless very useful: quite often discontinuities are not to be preserved for the expected outcome. In such a case it makes no sense to apply a smoothing technique that is influenced by discontinuities. In all other cases, TV flow is the better choice.

2.2 Structure Tensor

The structure tensor [FG87], also known as *second moment matrix*, is the next important tool described in this chapter. Its purpose is the estimation of orientation and the local analysis of structure in general. The structure tensor is successfully applied in many computer vision tasks, such as corner detection [FG87, HS88], optic flow estimation [BGW91, BWS05], and texture analysis [RS91, BGW91, MBLS01], yet it can also be useful in other fields of application outside the scope of image processing, e.g. for grid optimization in the scope of hyperbolic differential equations in computational physics [Tho99].

In order to integrate the structure information from a certain neighborhood, the classic structure tensor employs Gaussian convolution. The findings from the last section indicate that generally the accuracy of results can be improved, if Gaussian smoothing is replaced by discontinuity preserving nonlinear diffusion, in particular by TV flow. This idea can as well be transferred to the structure tensor [WB02, Bro02, BWBM04]. After a review of the conventional structure tensor based on Gaussian convolution, the *linear structure tensor*, thus a *nonlinear structure tensor* that is based on nonlinear diffusion and the diffusivities introduced in the previous section is proposed.

2.2.1 Linear Structure Tensor

The concept of the structure tensor is a consequence of the fact that one can only determine the orientation at a point reliably by considering also the data of its neighborhood. There are several disturbing artifacts like noise in an image which lead to estimates not having any relation to the real orientation at the respective point. Moreover, one is often not interested in the very local orientation at a specific point, but in the dominant orientation in a still local, but larger, neighborhood around this point.

The structure tensor therefore replaces the structure information of each pixel, which is described in a first order approximation by the gradient at that pixel, with the structure information of its surroundings weighted with a Gaussian window function. This comes down to the convolution of the structure data with a Gaussian kernel, i.e. Gaussian smoothing.

Note however, that the smoothing of gradients can lead to cancellation effects. Consider, for example, a thin line. At one side of the line there appears a positive gradient, while at the other side the gradient is negative. Smoothing the gradients will cause them to mutually cancel out. This is the reason why in the structure tensor, the gradient is considered in form of its outer product. The outer product turns the gradient vector ∇I into a symmetric positive semi-definite matrix, which I will refer to as the *unsmoothed structure tensor*

$$J_0 := \nabla I \nabla I^\top = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}. \tag{2.28}$$

Subscripts hereby denote partial derivatives. The structure tensor can be easily generalized from scalar-valued data to vector-valued data. As with the matrix representation it is possible to sum up gradient information, the structure information from all channels of a vector-valued image $\mathbf{I} = (I_1, ..., I_N)$ can be integrated by taking the sum of all matrices [Di 86]:

$$J_0 := \sum_{i=1}^{N} \nabla I_i \nabla I_i^\top. \tag{2.29}$$

The structure tensor for a certain neighborhood of scale ρ is then computed by Gaussian convolution applied to all components of J_0:

$$J_\rho = K_\rho * J_0. \tag{2.30}$$

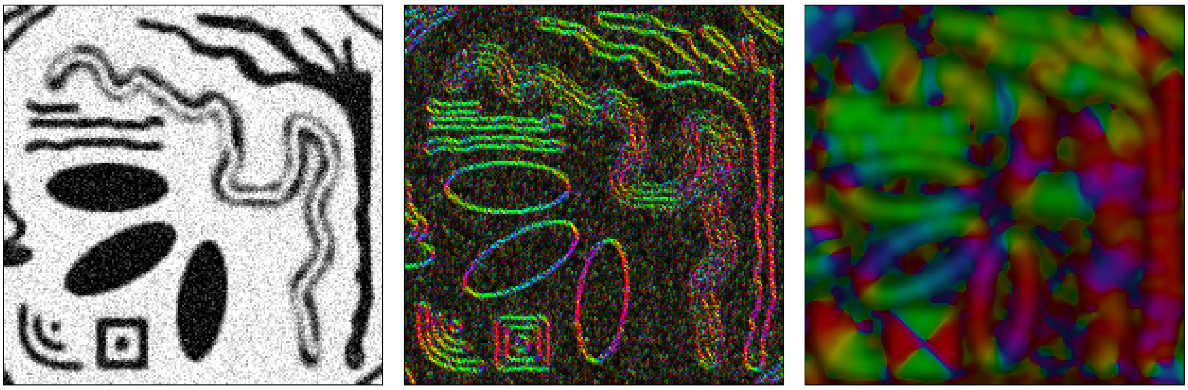


Fig. 2.22: LEFT: Synthetic image with Gaussian noise. CENTER: Unsmoothed structure tensor J_0. RIGHT: Linear structure tensor J_ρ with $\rho = 3$. Color indicates the orientation of the local structure.

The smoothing, i.e. the integration of neighborhood information, has two positive effects on the estimation of orientation. Firstly, it makes the structure tensor robust against noise or other artifacts, and therefore allows a more reliable estimation of orientation in real-world data. Secondly, it distributes the information about the orientation into the areas between edges. This closing of structures is a very important effect, as it allows to estimate the dominant orientation also at those points in the image where the gradient is close to zero.
In Fig. 2.22 one can see the effects of the smoothing on the structure tensor. The colored images are a way to show both the structure's dominant orientation, expressed by the hue, and its magnitude, expressed by the intensity.

Result of a least squares approach. The dominant orientation can be obtained from the structure tensor as the eigenvector to the largest eigenvalue. This shows up as a result of orientation estimation by means of a least squares approach (see e.g. [vdBvdW02]). Let $\mathbf{v}$ be the normalized orientation vector to be estimated. Then the total estimation error within the neighborhood defined by the Gaussian window function K_ρ is

$$\int_\Omega K_\rho \|\nabla I - (\nabla I^\top \mathbf{v})\mathbf{v}\|^2 dx = \int_\Omega K_\rho \left(\nabla I^\top \nabla I - (\mathbf{v}^\top (\nabla I \nabla I^\top)\mathbf{v}\right) dx \tag{2.31}$$

where Ω denotes the image domain and $(\nabla I^\top \mathbf{v})\mathbf{v}$ is the projection of ∇I on $\mathbf{v}$. Minimizing the error is equivalent to maximizing

$$\int_\Omega K_\rho \left((\mathbf{v}^\top (\nabla I \nabla I^\top)\mathbf{v}\right) dx \tag{2.32}$$

subject to the constraint $\mathbf{v}^\top \mathbf{v} = 1$. Since $\mathbf{v}$ is independent of x, this comes down to maximize

$$\mathbf{v}^\top \left(\int_\Omega K_\rho (\nabla I \nabla I^\top) dx\right) \mathbf{v} = \mathbf{v}^\top J_\rho \mathbf{v}. \tag{2.33}$$

So the orientation $\mathbf{v}$ that fits the data best in a least squares setting is the eigenvector corresponding to the largest eigenvalue of the structure tensor J_ρ.

Coherence information and corner detection. The consideration of a local neighborhood makes some additional information besides the orientation available: that of the homogeneity, or coherence, of the surrounding structure. Before smoothing it is not possible to distinguish a corner from an edge. Due to smoothing, however, the structure tensor is not only influenced by the orientation at a single position in the image, but by all orientations present in the local neighborhood. This means that the structure tensor is in general not singular anymore, so three cases can be distinguished when regarding the eigenvalues $\lambda_1 \geq \lambda_2$ of the matrix:

- $\lambda_1 \approx \lambda_2 \approx 0$: homogeneous areas, almost no structure present
- $\lambda_1 \gg 0, \lambda_2 \approx 0$: edges, one dominant orientation
- $\lambda_1 \gg 0, \lambda_2 \gg 0$: corners, structure with ambiguous orientation

Thus the coherence information makes the structure tensor directly applicable to corner detection [FG87].

2.2.2 Nonlinear Structure Tensor

Fig. 2.22 reveals that the Gaussian smoothing has not only positive effects but is also the cause for a negative property of the classic structure tensor: it leads to blurred and dislocated edges, which is a typical effect of Gaussian smoothing. From the last section we also know a remedy for this problem: the Gaussian convolution can be replaced by nonlinear diffusion, which preserves discontinuities. Intuitively, the preservation of discontinuities in the structure tensor can be understood as the implementation of a data-adaptive local neighborhood. This allows a much more accurate structure analysis, especially in the vicinity of discontinuities in the structure. In these areas, the local neighborhood induced by the Gaussian kernel integrates ambiguous structure information that actually does not belong together. Reducing the integration of further data as soon as a discontinuity is reached, avoids ambiguities and the structure can be estimated more precisely.

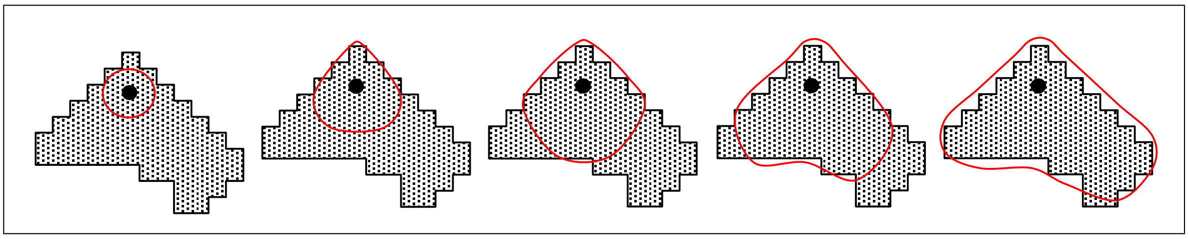


Fig. 2.23: Illustration of how, by an increasing amount of nonlinear diffusion, the local neighborhood is adapted to the region of similar data.

The principal concept of such a nonlinear structure tensor has been presented in [WB02] based on the matrix-valued diffusion scheme stated in (2.8). It has been examined more closely in [Bro02, BW02]. By means of the types of nonlinear diffusion that have been introduced in the last section, however, the nonlinear structure tensor can be further improved and may become more intuitive.

One of the most important requirements of the structure tensor is to distribute orientation information into the areas without gradient information. One should note that this requirement can be hard to fulfill with nonlinear diffusion, since nonlinear diffusion typically reduces the amount of smoothing at discontinuities and such discontinuities also appear in the passage to the gaps in the structure. Hence it is by no means clear whether this closing of structures works by applying nonlinear diffusion. However, the diffusivities introduced in (2.12)

$$g(|\nabla u|^2) = \frac{1}{|\nabla u|^p} \qquad p \in \mathbb{R}, p \geq 0.$$

which become unbounded for zero gradients, are a pretty good choice to ensure the closing in the nonlinear structure tensor. The structures in J_0 behave like an oscillatory signal (cf. Fig. 2.22) and the diffusion processes that correspond to these diffusivities remove mass from high peaks (maxima in the signal) and distribute it equally within the areas between the peaks (minima in the signal). The smaller the scale of the minima, the faster they are filled and the faster they meet the surrounding peaks. Thus let the nonlinear structure tensor J_t be defined as a solution of the diffusion equation

$$\partial_t J_{ij} = \operatorname{div}\left(g\left(\sum_{k,l} |\nabla J_{kl}|^2 \right) \nabla J_{ij} \right) \tag{2.34}$$

based on the matrix-valued diffusion scheme from (2.8), with the matrix J_0 as initial condition and with g being one of the diffusivities from (2.12).

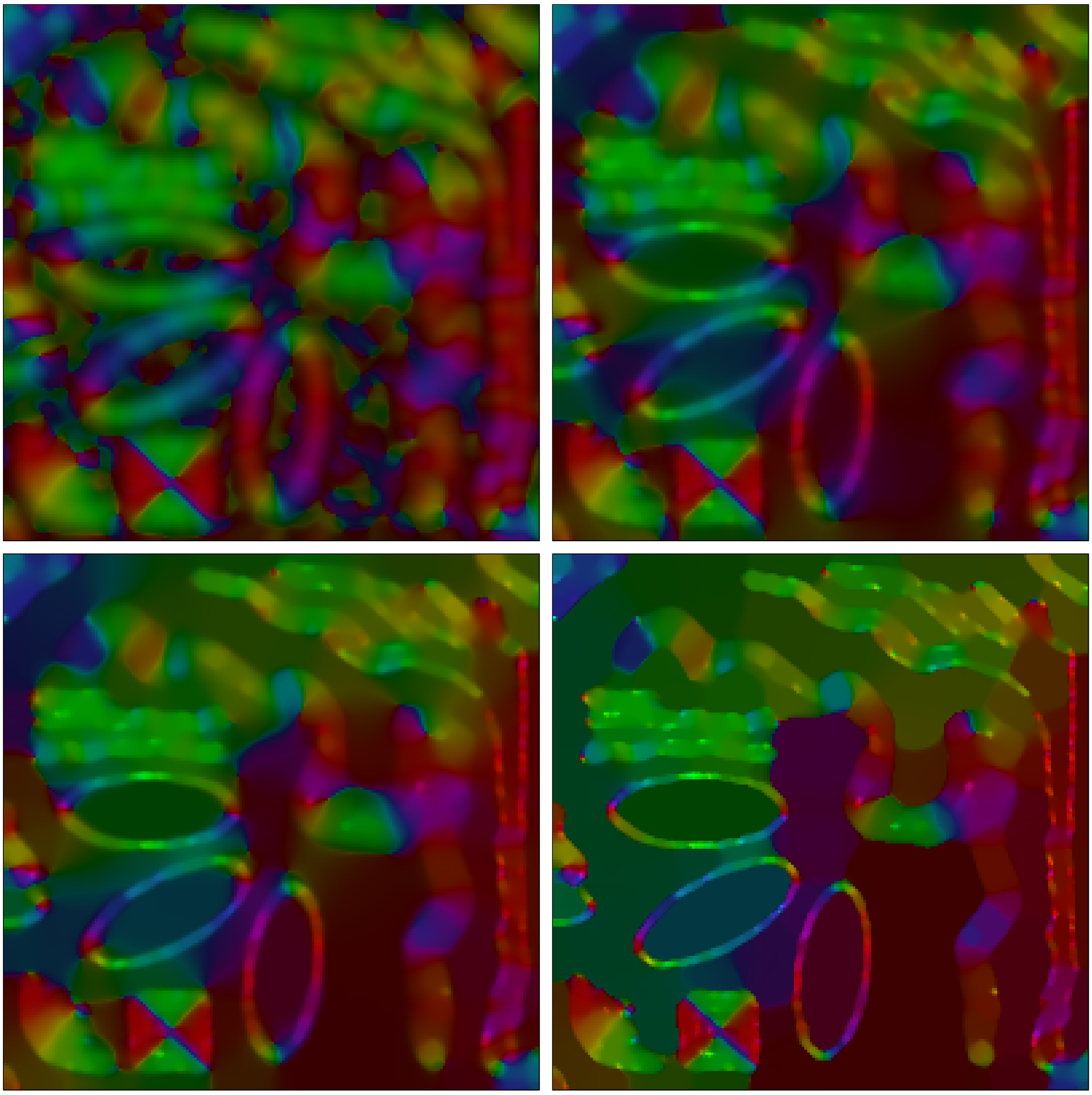


Fig. 2.24: Nonlinear structure tensor for different choices of p.
Top Left: $p = 0$, $t = 4.5$ (classic structure tensor). Top Right: $p = 0.8$, $t = 800$.
Bottom Left: $p = 1$, $t = 3200$. Bottom Right: $p = 1.2$, $t = 12000$.

A nice side effect of using this family of diffusivities is the fact that the conventional structure tensor is included in the nonlinear structure tensor as a special case when p is set to 0. This means the nonlinear structure tensor generalizes the classic structure tensor by adding one degree of freedom, that is the amount of discontinuity preservation steered by means of p. In the scope of the structure tensor, this corresponds to the rigidity with which the extension of the local neighborhood is reduced at discontinuities in the data. Small p correspond to a local neighborhood that is still close to a Gaussian window, while increasing p adapt this neighborhood more and more to the data, in a way described by the behavior of the underlying nonlinear diffusion process.

Experiments. Fig. 2.24 shows the nonlinear structure tensor for different choices of p. One can observe very well that the blurring artifacts known from the classic structure tensor are greatly reduced by the choice of larger p. Nevertheless, it is still ensured that structures of a certain scale are closed.

Like the classic linear structure tensor, also the nonlinear structure tensor creates a scale space with the scale parameter t. In case of the nonlinear structure tensor, the scale parameter mainly influences the scale of the structures that are closed. An example is depicted in Fig. 2.25.

In Chapter 4 the nonlinear structure tensor will be applied to texture discrimination and optic flow estimation. In the experiments there, one will be able to see that the higher accuracy of the nonlinear structure tensor really improves the quality of the results.

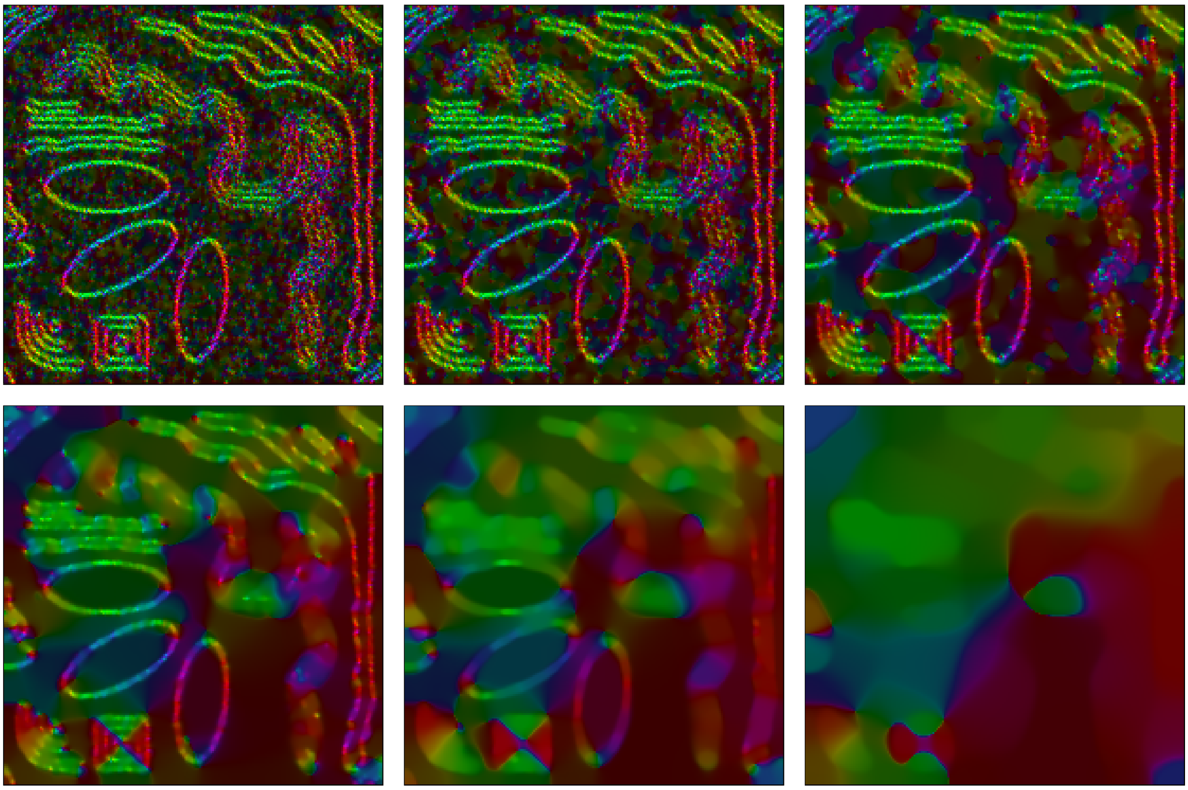


Fig. 2.25: Scale space evolution of the nonlinear structure tensor ($p = 1$).
Top row, from Left to Right: $t = 250$. $t = 500$. $t = 1000$.
Bottom row, from Left to Right: $t = 2000$. $t = 4000$. $t = 8000$.

Relations to other data-adaptive structure tensors. The main difference between the nonlinear and the linear structure tensor lies in the capability of the nonlinear structure tensor to adapt the neighborhood to the data. There are alternative approaches to data-adaptive structure tensors in the literature. Nagel and Gehrke [NG98] have introduced the so-called *gray value local structure tensor*, which they have designed for its use in spatio-temporal optic flow estimation. Instead of using a fixed isotropic Gaussian kernel K_ρ for smoothing the structure tensor, like in the linear structure tensor, a data-adaptive Gaussian

$$G(x) = \frac{1}{\sqrt{(2\pi)^3|\Sigma(x)|}} e^{-\frac{1}{2}x^\top \Sigma(x)^{-1}x} \tag{2.35}$$

which is parameterized by the covariance matrix $\Sigma(x)$, is employed. The covariance matrix is locally adapted to the image by setting

$$\Sigma(x) = U(x) \begin{pmatrix} \sigma_{min} + \frac{\sigma_{max}^2}{1+\sigma_{max}^2\lambda_1(x)} & 0 & 0 \\ 0 & \sigma_{min} + \frac{\sigma_{max}^2}{1+\sigma_{max}^2\lambda_2(x)} & 0 \\ 0 & 0 & \sigma_{min} + \frac{\sigma_{max}^2}{1+\sigma_{max}^2\lambda_3(x)} \end{pmatrix} U^\top(x) \tag{2.36}$$

where $\lambda_i(x), i \in \{1,2,3\}$ are the eigenvalues of the evolving structure tensor and U holds its eigenvectors. The covariance matrix is initialized by a diagonal matrix corresponding to an isotropic Gaussian window. The parameters σ_{min} and σ_{max} are for restricting the anisotropy and the size of the Gaussian. This concept of using a data-adaptive Gaussian for the convolution with the structure tensor has been further investigated in the works of Middendorf and Nagel [MN01, MN02].

Fig. 2.26 visualizes the differences between this approach and the linear as well as the nonlinear structure tensor. The linear structure tensor uses a fixed isotropic Gaussian kernel for smoothing the data, thus it is not data-adaptive at all. The method proposed by Nagel and Gehrke parameterizes the neighborhood by an anisotropic Gaussian and adapts the parameters locally to the data. Although this approach is more precise than the linear structure tensor, one can see that in many situations the Gaussian cannot fully cover the region of interest without also integrating ambiguous information. The iterative diffusion process involved in the nonlinear structure tensor is much more flexible and can therefore cover a neighborhood with arbitrary shape.

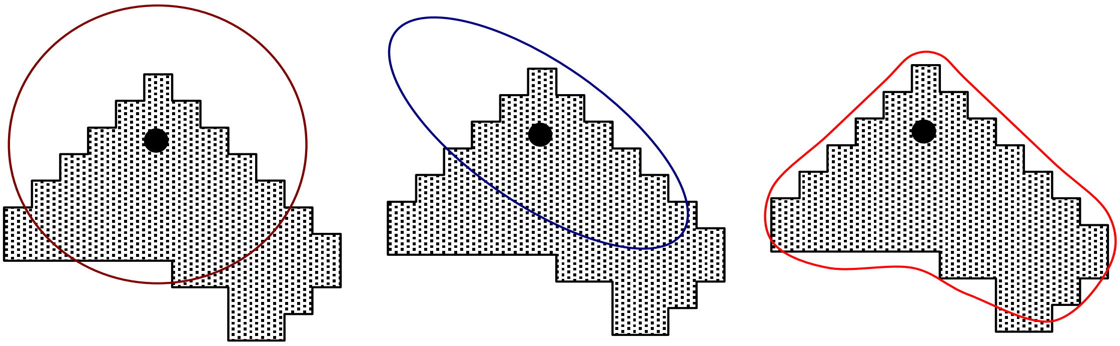


Fig. 2.26: LEFT: Neighborhood of a non-adaptive isotropic Gaussian.
CENTER: Neighborhood of a data-adaptive anisotropic Gaussian.
RIGHT: Neighborhood obtained with a nonlinear diffusion process (cf. Fig. 2.23).

Another nonlinear approach to a data-adaptive structure tensor has been introduced in [vdBvdW02] where the authors propose robust statistics to estimate the dominant orientation. In robust statistics one replaces the error norm $\Psi(s^2) = s^2$ by a robust function, like for instance $\Psi(s^2) = 1 - e^{-\frac{s^2}{2\lambda^2}}$. The goal of such a robust function is to decrease the influence of so-called outliers, i.e. single values that do not fit to the current estimate due to noise or because they belong to another entity and therefore cause a large error when using the squared error norm.

As stated above, the least squares estimate of the dominant orientation is the eigenvector to the largest eigenvalue of the classic structure tensor (cf. (2.31) - (2.33)). Using a robust function instead of the square function in (2.31) leads again to an eigenvalue problem[3]

$$J_{\Psi,\rho}(\mathbf{v})\mathbf{v} = \lambda \mathbf{v} \tag{2.37}$$

but now with a structure tensor

$$\begin{aligned} J_{\Psi,\rho}(\mathbf{v}) &= \int_\Omega \Psi'(\nabla I^\top \nabla I - \mathbf{v}^\top (\nabla I \nabla I^\top)\mathbf{v})\; K_\rho\; \nabla I \nabla I^\top\; dx \\ &= \frac{1}{2\lambda^2} \int_\Omega e^{-\frac{\nabla I^\top \nabla I - \mathbf{v}^\top (\nabla I \nabla I^\top)\mathbf{v}}{2\lambda^2}}\; K_\rho\; \nabla I \nabla I^\top\; dx \end{aligned} \tag{2.38}$$

that depends on the estimated orientation $\mathbf{v}$. One can see that dependent on how well the values fit to the currently estimated orientation, their influence is decreased. This is similar to the concept of the nonlinear structure tensor, where the further expansion of the local neighborhood is reduced if the new values do not fit well to the values of the current neighborhood. Note that the weighting function $\Psi'(s^2)$ in (2.38) is, besides a constant factor, one of the Perona-Malik diffusivities used also for nonlinear diffusion (cf. Eq. 2.5). Both the nonlinear structure tensor and the structure tensor based on robust statistics make the integration of further data dependent on whether it fits to the already gathered data. The difference between both approaches is that the nonlinear structure tensor applies this selection process in order to determine the local neighborhood and then uses a simple least squares approach within this neighborhood, while the structure tensor based on robust statistics first gathers the data from the simple fixed Gaussian neighborhood K_ρ and applies the nonlinear weighting process afterwards. For an experimental validation of these differences see also [BvdBL+05].

[3] For further details see [vdBvdW02].

2.3 Variational Methods

Besides nonlinear diffusion and the nonlinear structure tensor there is another very important technique based on partial differential equations, that is the concept of variational methods. In a variational approach one formulates some model assumptions $A_1, ..., A_m$ in terms of an *energy functional*

$$E(u_1(\mathbf{x}), ..., u_n(\mathbf{x})) = \int_\Omega (A_1 + ... + A_m)\,\mathbf{dx} \qquad (2.39)$$

and tries to find those functions $u_1, ..., u_n$ that minimize the energy, possibly by respecting additional constraints. Such an approach has various good properties. Firstly, it is necessary to quantify the model assumptions by so-called *penalizer terms*. Each penalizer induces a high energy for those cases where the model assumption is not fulfilled and a low energy otherwise. Due to the necessity to quantify the assumptions, one is forced to clearly think about which assumptions are included in the model. This avoids hidden assumptions and yields a sound description of the problem that has to be solved. It is relatively easy to read the whole problem statement from the single formula of the energy functional.

Nonetheless variational methods are very general. One can formulate rather enhanced problem statements in terms of an energy functional, and the choice of how to design the penalizers yields a high flexibility. Due to this generality, variational methods are also applicable to higher level vision tasks, where many other image processing techniques are not suitable to handle the problem anymore.

Further on, with variational methods one gets access to well-founded mathematical techniques and numerics. The theory of the calculus of variations[4] provides a way how to minimize the energy functional. It leads to the so-called *Euler-Lagrange equations*, which have to be satisfied in a minimum. The Euler-Lagrange equations are partial differential equations, what draws the connection to the diffusion methods discussed earlier in this chapter. For sufficiently simple energy functionals, these Euler-Lagrange equations lead to a linear system of equations, which can be solved by well-founded and optimized numerical methods. Thus for many energy functionals not only the modelling process is very sound, one can even be sure to obtain really the optimum solution that corresponds to the model.

2.3.1 Variational Denoising with Quadratic Penalizers

Consider the following energy functional for image denoising in order to illustrate the concept of variational methods [Tik63]:

$$E(u) = \int_\Omega \Big(\underbrace{(u - I)^2}_{\text{Data}} + \alpha \underbrace{|\nabla u|^2}_{\text{Smoothness}} \Big)\,\mathbf{dx} \qquad (2.40)$$

Although this is a very simple energy functional, the same configuration can be found in many other models in image processing. The first term, called the data term, penalizes deviations of the result from the original image I. It expresses the assumption that the result should be close to the data. The second term, called the smoothness term, additionally penalizes deviations from smoothness. The underlying assumption thereby is that the result should be smooth. In case of noise or discontinuities in the image, the two assumptions are certainly contradictive and the optimum solution must be a compromise between both assumptions. The relative importance of the two assumptions is steered by the smoothness parameter α.

[4] This is where the term *variational method* comes from.

For larger α the smoothness assumption becomes more important, so the optimum solution u will be smoother. The other way round, if α is set to 0, the optimum solution will be to completely fulfill the data term, so the result will be the image I itself.

The answer to the question of how the optimum solution can be computed is given by the theory of the calculus of variations [CH53, Els61]. The minimizing function u must satisfy the Euler-Lagrange equation

$$(u - I) - \alpha \operatorname{div}(\nabla u) = 0 \tag{2.41}$$

with reflecting boundary conditions $\partial_x u = 0$ and $\partial_y u = 0$. After a suitable discretization of the divergence expression (cf. Section 2.1.3) this leads to a large, but sparse, linear system of equations which can be solved by means of an iterative solver like Gauss-Seidel or SOR [You71]. In this thesis, the SOR method is preferred, because it leads to much faster convergence than the Gauss-Seidel method and is still sufficiently easy to implement[5]. The iteration scheme that solves for the minimizer of (2.40) is

$$u_i^{k+1} = (1-\omega)\, u_i^k + \omega\, \frac{\sum_{j\in\mathcal{N}^-(i)} u_j^{k+1} + \sum_{j\in\mathcal{N}^+(i)} u_j^k - \frac{I_i}{\alpha}}{|\mathcal{N}^-(i)| + |\mathcal{N}^+(i)| + \frac{1}{\alpha}} \tag{2.42}$$

where $\mathcal{N}^-(i)$ denotes the neighbors j of i with $j < i$, and $\mathcal{N}^+(i)$ the neighbors j of i with $j > i$. The scheme converges for the over-relaxation parameter $\omega \in (0, 2)$. The choice of ω where the scheme shows the best convergence depends on the linear system that has to be solved. For the example shown here, values close to 2 perform best. For $\omega = 1$ one obtains the Gauss-Seidel method. It should be noted that with subtle multi-grid techniques it is possible to solve such a kind of linear system even faster than with SOR. For details see [BWF$^+$03, BWF$^+$05].

As an alternative one can also find the solution by means of a gradient descent

$$\partial_t u = \operatorname{div}(\nabla u) - \frac{1}{\alpha}(u - I) \tag{2.43}$$

which means in the case here that one searches the steady-state of a diffusion process with a bias term. This shows the relations of variational denoising methods to diffusion processes, which will be discussed in more detail later in Section 2.3.4. Searching the minimizer of (2.40) with an explicit gradient descent leads to a much slower convergence than the iteration scheme in (2.42). In the case of more complicated energy functionals that do not lead to linear systems of equations, however, the gradient descent is often the only possibility to get a solution at all.

2.3.2 Non-quadratic Penalizers

The energy functional in (2.40) uses a quadratic penalizer $\Psi(s^2) = s^2$ for both the data term and the smoothness term. Quadratic penalizers are very convenient, since they lead to linear Euler-Lagrange equations. However, they often do not describe the model one actually wants to formulate. Quadratic penalizers give much influence to those points that are far away from the assumptions. Since each deviation from the assumption leads to a squared error, the optimum solution is adapted in such a way that it is closer to these outliers. So the model does simply not *accept* that there can be outliers in one of the assumptions. A model description which accepts outliers has to penalize them less severely [Hub81, HRRS86].

[5] For the SOR method to converge, the system matrix has to be strictly positive or negative definite, which is the case here.

A typical penalizer function that fulfills this requirement is $\Psi(s^2) = \sqrt{s^2} = |s|$. If we replace the smoothness term in (2.40) by $|\nabla u|$, we allow some outliers in smoothness, i.e. we allow discontinuities in the final result. The energy functional using this penalizer

$$E(u) = \int_{\Omega} \left((u - I)^2 + 2\alpha |\nabla u| \right) \, \mathbf{dx} \tag{2.44}$$

corresponds to TV regularization initially introduced by Rudin et al. [ROF92]. It has been proposed in this form by Acar and Vogel in [AV94]. The Euler-Lagrange equation of this energy is

$$(u - I) - \alpha \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) = 0 \tag{2.45}$$

so by minimizing the energy one searches the steady-state of TV flow with a bias term:

$$\partial_t u = \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) - \frac{1}{\alpha}(u - I). \tag{2.46}$$

We will see later in Section 2.3.4 that in the discrete 1-D setting of this special case, it is even not necessary to search for a steady-state of the diffusion equation with bias term, since also TV flow without bias term yields the minimizer of TV regularization when the diffusion time t is set equal to α.

For the moment let us focus on the nonlinearity of the system of equations that emerges after spatial discretization of (2.45) [6]. The divergence expression in (2.45) contains $\Psi'(|\nabla u|^2) = 1/|\nabla u|$, which depends on u. This is in contrast to the *linear* system of equations obtained with the quadratic penalizer where Ψ' was a constant. Thus the problem to be solved now is considerably more complicated.

Aside from the immediate, but slowly converging, gradient descent in (2.46), there is a more efficient way to solve such a nonlinear system of equations[7]. One can transform the nonlinear system of equations into a sequence of linear systems by a so-called *fixed point iteration scheme* [Cia78, WHS$^+$01]. One starts with some initialization u^0 at iteration $k = 0$ [8]. Keeping u^k fixed for computing $\Psi'(|\nabla u^k|^2)$, we obtain again a linear system of equations. Solving this linear system leads to u^{k+1} and an update $\Psi'(|\nabla u^{k+1}|^2)$ can be computed[9].

According to Kačanov [Cia78], such a fixed point iteration scheme converges under the condition that the linear systems are solved correctly. Since for 2-D data structures like images an iterative solver is necessary to efficiently solve the linear systems, this condition that the solver has to fully converge induces rather high computational costs. In practice it is much more efficient to do only a few iterations on the linear system and then update the fixed point again. Experimental validation shows that such a procedure leads to a good convergence as well, yet there is no proof of convergence yet available and one has to set the number of inner iterations somehow arbitrarily. This is a small deviation from the sound theory of variational methods for the sake of a more sophisticated model and a higher efficiency.

[6] The discretization is the same as described in 2.1.3.

[7] From Section 2.1.3 we know that there are also fast implicit methods for solving this parabolic problem, yet they are also not immediate anymore, and the methods currently available still lead to a slower convergence than the method described here. See also the thesis of Bruhn for a much deeper insight into efficient numerical schemes for parabolic and elliptic problems [Bru05].

[8] In the case of image denoising $u^0 = I$ seems most reasonable.

[9] An alternative minimization algorithm with a proof of convergence for the special case of TV regularization can be found in [Cha04].

2.3.3 Multi-modal Functionals

We implicitly assumed so far that the considered energy functionals have a unique solution. However, this can only be ensured by putting some constraints on the functional and its penalizer function Ψ. It could be proven that if $\Psi(s^2)$ in the regularization functional

$$E(u) = \int_\Omega \left((u-I)^2 + \alpha\Psi(|\nabla u|^2|)\right) \mathbf{dx} \tag{2.47}$$

is increasing and differentiable, and it is convex in s, then there exists a unique minimizer that is twice differentiable and depends continuously on I [Sch94c]. Note that the penalizer used in (2.44) is just at the limit to convexity and it is not differentiable in 0. In order to guarantee a unique minimum, it is therefore necessary to apply an ϵ-regularization on the penalizer (cf. Section 2.1.3). With the penalizer function $\Psi(s^2) = \sqrt{s^2+\epsilon^2}$ with $\epsilon > 0$ a unique solution can be ensured.

However, what happens if the penalizer is non-convex? In such a case, the energy functional can have multiple minima. Consequently, the local minimization strategies investigated so far, converge depending on the initialization to one of these local minima. This need not necessarily be the global minimum.
Non-convex penalizer functions are not the only source of so-called *multi-modal* functionals, i.e. energy functionals which have more than one local minimizer. The problem of local minima becomes even more severe as soon as more complex models based on nonlinear constraints or the simultaneous optimization of two or more functions are considered. Such models are hardly to avoid as soon as one considers computer vision problems, where the model has to abstract more from the given image data[10].

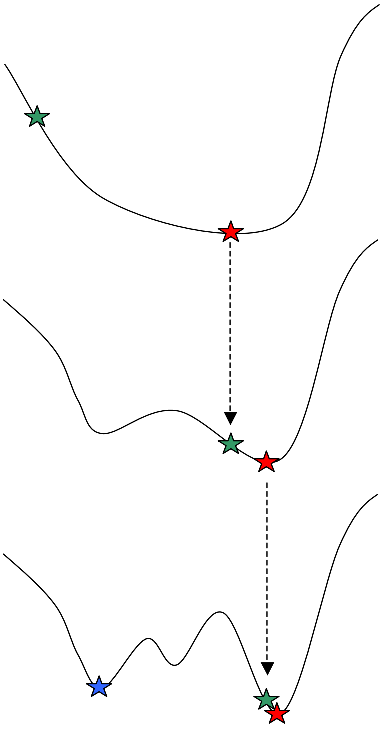


Fig. 2.27: Illustration of continuation methods.

There is no general compliance on how do deal with the problem of local minima. In some cases it is sufficient to end up in a local minimum that depends on the initialization. This is often the case in medical applications of segmentation, where a doctor can give a rough outline of the object to be segmented and the optimization process yields the next best segmentation based on this initialization. In other cases, however, it is favorable to be independent from the initialization. So how to avoid being trapped by a local minimum, when there exists a much better global optimum? A popular solution is the use of so-called *continuation methods* or *graduated non-convexity (GNC) methods* [BZ87]. The basic idea is to transform the complicated original functional to a simplified smoother functional, where a unique minimum exists. See Fig. 2.27 for illustration. Using the heuristic that the minimum of the simplified functional is a good initialization for solving a less simplified version, it becomes possible to reach the global optimum of the original functional step by step. The green spot in Fig. 2.27 marks the initialization at the respective level of simplification, and the red spot marks the

[10] After discussing image denoising in this section, a more complex model in the field of optic flow computation will be regarded in Section 4.2. Later in Chapter 5, a variational model for image segmentation will be investigated.

minimum reached with this initialization. One can see that the continuation method in this example makes the optimization indeed independent from the initialization (cf. the blue spot that marks the optimum found without the continuation method).

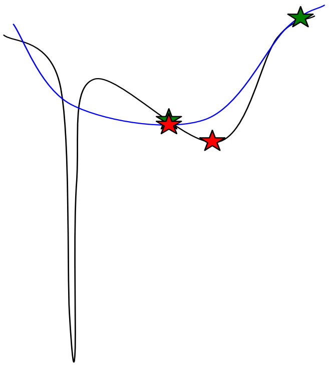


Fig. 2.28: Continuation method fails.

However, there are also counterexamples where this procedure must fail. One such example is shown in Fig. 2.28. If finding the global optimum is literally looking for a needle in a haystack, then also a simplified functional does not help. The only solution so far, is to avoid models that might lead to such unpleasant functionals. Unfortunately, it is impossible to visualize the huge search space of energy functionals, yet one can imagine that for example the multiplication of assumptions

$$E(u) = \int_\Omega A_1(u) \cdot A_2(u) \, \mathbf{dx} \tag{2.48}$$

yields very nasty optimization problems.

There is certainly much room for research when it comes to the optimization of increasingly complex energy functionals. So far, there is quite a range of functionals where the optimization at least works experimentally, though there is still a proof of convergence missing. Such examples will be shown in Section 4.2 and Chapter 5. As emphasized before, a variational approach ensures a sound modelling with all model assumptions clearly stated. However, heuristics still have to be applied for many optimization problems and cannot guarantee the global optimum according to the model. Getting more theoretical insight into the optimization procedures is therefore important to ensure the advantages of variational methods also in the case of more complex problems.

2.3.4 Variational Denoising and its Relations to Nonlinear Diffusion

We now return to the case of variational denoising, which is already very well understood. From the Euler-Lagrange equations and the gradient descent equations (2.43) and (2.46) it has already been possible to observe a strong relation between variational denoising and nonlinear diffusion described in Section 2.1. Indeed the task of both techniques is the same: the smoothing of image data. While nonlinear diffusion models this smoothing as a physical process, the variational approach expresses the task by a smoothness assumption. Both approaches end up in very similar partial differential equations [KDA97].

Fully implicit time discretization of a diffusion equation. Scherzer and Weickert [SW00] substantiated the relations between nonlinear diffusion, given by the diffusion equation

$$\partial_t u = \text{div}\ (g(|\nabla u|)\nabla u) \tag{2.49}$$

with initial condition $u = I$, and variational denoising, given by the energy functional

$$E(u) = \int_\Omega \left((u-I)^2 + \alpha\Psi(|\nabla u|^2|)\right) \, \mathbf{dx}. \tag{2.50}$$

The minimizer of the energy functional must satisfy the Euler-Lagrange equation

$$(u-I) - \alpha \ \text{div} \ \left(\Psi'(|\nabla u|^2)\nabla u\right) = 0. \tag{2.51}$$

After rewriting this equation as

$$\frac{u-I}{\alpha} = \text{div}\,\left(\Psi'(|\nabla u|^2)\nabla u\right) \tag{2.52}$$

it becomes obvious that it may be regarded as a fully implicit time discretization of the diffusion equation (2.49) with diffusivity $g(s^2) = \Psi'(s^2)$, initial value I, and stopping time $t = \alpha$. On the basis of this relation, Scherzer and Weickert were able to prove in [SW00] that many theoretical properties of nonlinear diffusion carry over to variational denoising, like e.g. the maximum-minimum principle and convergence to the average gray level for $\alpha \to \infty$.
From the fact that (2.52) is a fully implicit time discretization of (2.49) it can be expected that the minimizer of (2.50) approximates the diffusion filter. Indeed the minimizer shown in Fig. 2.29 on the left looks very much like the result of the corresponding homogeneous diffusion depicted on the right, though the result of the diffusion process is for convex Ψ a bit smoother.

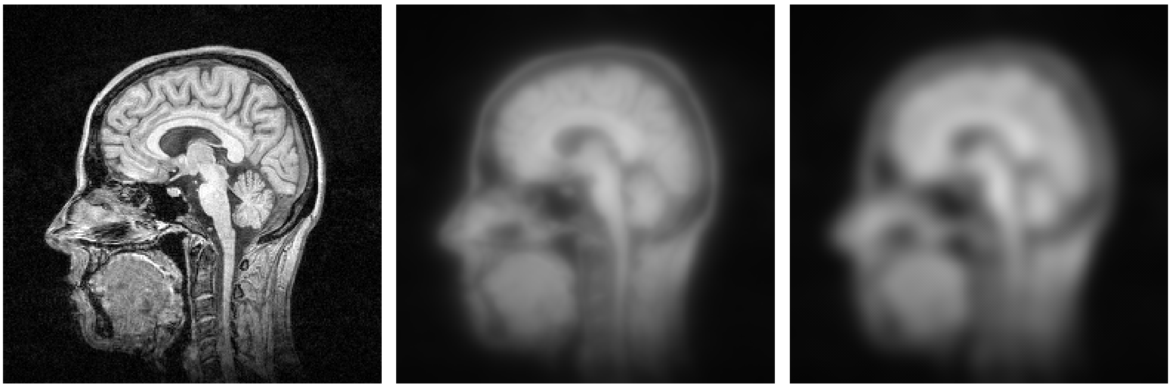


Fig. 2.29: Comparison of variational denoising and diffusion. LEFT: Original image. CENTER: Tikhonov regularization with $\alpha = 20$. RIGHT: Homogeneous diffusion with $t = 20$.

Equivalence of diffusion and variational denoising. While the above-mentioned relation establishes an *approximation* between the corresponding processes, one may be interested in results where one can establish even *equivalence* between a diffusion filter and a variational method. For simplicity we consider only the one-dimensional versions of the filtering techniques. Some of the results also carry over to the higher dimensional case.

Nielsen et al. [NFD97] have shown that the solution of the linear diffusion filter [Iij62, WII99]

$$\begin{aligned} u_t &= u_{xx} \\ u(x,0) &= I(x) \end{aligned} \tag{2.53}$$

at time $t = \alpha$ may be regarded as the exact minimizer of an energy functional with an infinite number of penalizing terms of arbitrarily high order:

$$E(u) = \int\limits_{\mathbb{R}} \left((u-I)^2 + \sum_{k=1}^{\infty} \frac{\alpha^k}{k!} \left(\frac{d^k u}{dx^k} \right)^2 \right)\, dx. \tag{2.54}$$

An equivalent result has also been obtained earlier by Yuille and Grzywacz in the context of visual motion perception [YG88a, YG88b].

Another linear PDE-based filter is given by the pseudodifferential equation [FS01, DFdGtH04]

$$\begin{aligned} u_t &= -\sqrt{-\frac{\partial^2}{\partial x^2}}\, u \\ u(x,0) &= I(x). \end{aligned} \tag{2.55}$$

Duits et al. [DFdGtH04] have shown that this so-called *Poisson scale-space* may be regarded as the exact minimizer of

$$E(u) = \int\limits_{\mathbb{R}} \left((I-u)^2 + \sum_{k=1}^{\infty} \frac{\alpha^k}{k!} \left(\left(-\frac{d^2}{dx^2} \right)^{k/4} u \right)^2 \right) dx. \tag{2.56}$$

Obviously, people have been able to derive energy functionals that are equivalent to the evolution equation only in the *linear* case, so far. Moreover, these functionals are relatively complicated, since they involve an infinite number of regularizing terms.

Surprisingly, it turns out that there exists a *nonlinear* framework, in which there is an equivalence between diffusion filtering and regularization, that has a significantly *simpler* structure than equivalences in the linear case. This framework is given by TV regularization, defined by (2.44), and TV flow, described in Section 2.1.1.

In Section 2.1.1 it has been exposed that in the spatially discrete 1-D case it is possible to derive an analytical solution of TV flow by means of a couple of evolution rules. Analytical results have also been derived for continuous TV regularization by Strong [Str97] and for the taut-string algorithm in statistics by Mammen and van de Geer [MvdG97]. In accordance with these findings evolution rules for space-discrete TV regularization are proven in Appendix A. The structure of this proof is in analogy with the proof for TV flow.
Since the evolution rules derived for TV regularization turn out to be exactly the same rules as those for TV flow, this immediately establishes equivalence between both filtering methods in the discrete 1-D setting. The same result has independently been shown later on by Pollak et al. [PWH05].

Comparing numerical schemes for TV flow to numerics for TV regularization. The equivalence between TV flow and TV regularization is quite useful, not only for putting oil on troubled waters of advocators of both paradigms, but also for transferring theoretical results or even numerics from diffusion to variational denoising and vice versa.
In Section 2.1.3 several numerical schemes for TV flow have been compared with regard to their accuracy. The equivalence between TV flow and TV regularization allows to add also a numerical scheme for TV regularization to this comparison. For transferring the results to higher dimensions, however, it has to be noted that equivalence is solely proven in 1-D so far.

The Euler-Lagrange equation of the TV regularization functional yields, after discretization, a nonlinear system of equations that can be transformed into a sequence of linear systems by means of fixed point iterations (cf. Section 2.3.2). As we compare only 1-D schemes[11], the Thomas algorithm [Tho49, WtHV98] can be applied in order to solve the linear systems. The critical parameter with regard to the accuracy is the number of fixed point iterations. In Fig. 2.32 one can see the L_2-error in comparison to the errors of the various diffusion schemes in dependence on the level of accuracy. The number of fixed point iterations for each level of accuracy in the diagram has been set to the number of iterations needed for the AOS scheme

[11] Only in 1-D the correct solution is available, thanks to the analytic solution, and allows a fair comparison.

at the same level. It can be observed that for very fast but inaccurate processing, the AOS scheme is more appropriate. Fig. 2.31 reveals that the output is too close to the original signal, if the number of iterations is too small. On the other hand, after a certain number of iterations the regularization approach is more accurate and can even reach the accuracy of the two-pixel scheme. A comparison of the convergence of three different TV regularization algorithms can be found in [Vog02].

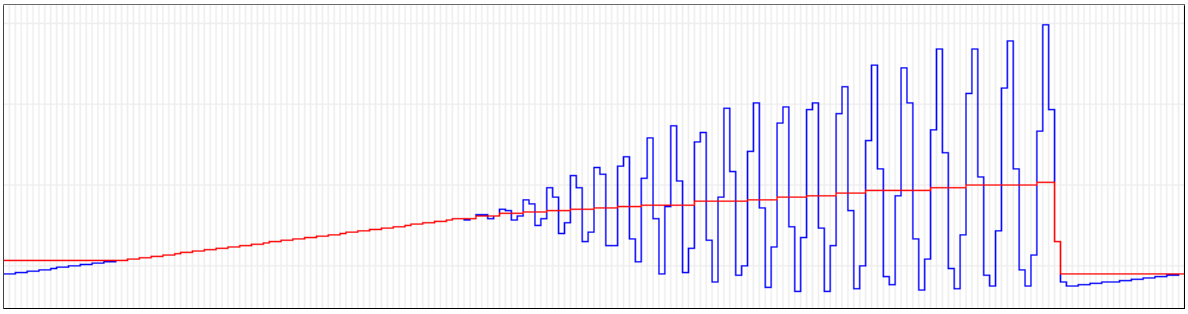


Fig. 2.30: Blue Line: Input signal from Fig. 2.18.
Red Line: Exact solution of TV flow after $t = 1.75$.

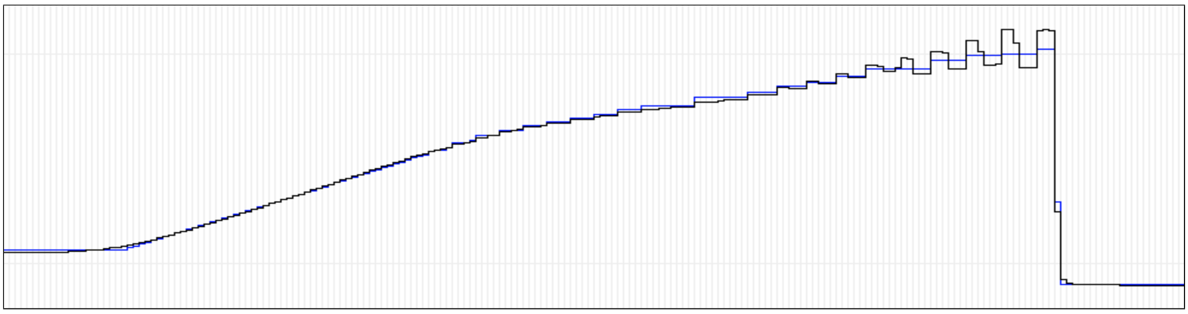


Fig. 2.31: Evolution result for the signal in Fig. 2.30 after diffusion time $t = 1.75$.
Blue Line: Exact solution of TV flow.
Black Line: TV regularization with $\epsilon = 0.001$ and 70 iterations.

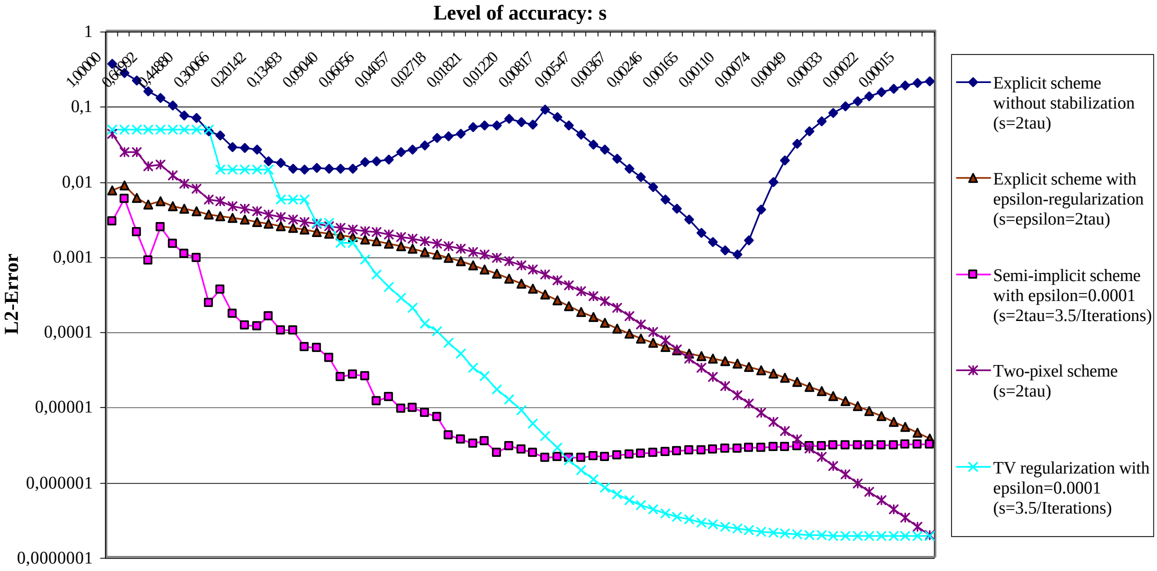


Fig. 2.32: Comparison of the average L_2-error with various numerical schemes for TV flow including implementation via TV regularization. The underlying signal is depicted in Fig. 2.30. Diffusion time $t = \alpha = 1.75$. See also Fig. 2.18-2.21.

3 Image Processing Techniques in Hyperbolic Conservation Laws

Faith is a poor substitute for logic and reason.

In the previous chapter, special nonlinear diffusion filters have been presented. The reader may recall the term of diffusion actually from another subject in science: physics. Indeed the motivation of many diffusion equations has been adopted from physics. It is solely the reinterpretation of mass as the image gray value that allows for the application of the same process in image processing, which is, in principle, a very different field of research with different objectives.

Apart from diffusion processes, there are other concepts from physics that have found their application in image processing. Starting with Rudin's Ph.D. thesis in 1987 [Rud87], many ideas from computational fluid dynamics and the numerics of hyperbolic conservation laws have entered the field of image processing. Problems of fluid dynamics and hyperbolic conservation laws involve the formation of shocks. On the image processing side, image edges carry important information and may be regarded as shocks as well. For instance, a common problem in image processing is concerned with blurred edges. So it is a natural idea to apply shock-enhancing concepts from computational fluid dynamics to enhance edges in images [OR90, OR91].
In another application, travelling shock waves are regarded as evolving object contours. Tracking the level lines of a function allows for the modelling of morphological operators like dilation and erosion within the so-called *level set framework* [DT79, OS88]. Going a step further and embedding a curve as the zero-level line into an auxiliary function, the so-called *embedding function* or *level set function*, leads to implicit active contours [CCCD93, MSV95, CKS95, KKO^{+}95] which perform the evolution of an object contour within the level set framework. This technique will be part of the segmentation method introduced in Chapter 5.

While numerical ideas for hyperbolic conservation laws had undoubtedly a strong impact on modern image analysis, fertilization in the inverse direction – where image processing methods are applied to improve the numerics of hyperbolic conservation laws – have started only recently: in [GMS02, GS02] different variants of numerical schemes are proposed that combine the second order Lax-Wendroff scheme with anisotropic diffusion filtering as it has been developed in image processing [Wei98a]. A related technique has been published in [Wei02]. Both strategies start with a hyperbolic scheme that gives sharp shock resolution, but suffers from oscillations in the shock areas. Anisotropic diffusion regards such oscillations as noise at edges that can be removed by smoothing along the edge.
The other way round, one could as well be interested in enhancing a monotone hyperbolic scheme which prevents oscillations but in return introduces blurring effects that reduce the shock resolution. An adequate inverse diffusion filter can reduce such blurring effects.

The concept to use image enhancement methods in order to improve numerical schemes in fluid dynamics is the topic of the present chapter. Research in this field has been in cooperation with Michael Breuß and Andrea Bürgel from the Technical University of Braunschweig.

Hyperbolic conservation laws. Many problems in fluid dynamics satisfy hyperbolic conservation laws of type

$$\partial_t u + \partial_x (f(u)) = 0 \tag{3.1}$$

where $u := u(x,t)$ is a scalar-valued function of a one-dimensional space variable x and time t, and where the flux function $f(u)$ is supposed to satisfy $f'(u) \geq 0$. Furthermore, reflecting boundary conditions are assumed.

Depending on the flux function $f(u)$ and the complexity of the initial condition, there is often no analytic solution for differential equations of such type. One is therefore interested in finding numerical solutions by approximating the differential equation with a numerical scheme.

In this chapter, three different prototypes for flux functions $f(u)$ are considered:

- *Linear advection*

  $$\partial_t u + \partial_x u = 0 \tag{3.2}$$

  This is a simple linear partial differential equation which just shifts the mass in positive spatial direction with speed 1.

- *Inviscid Burgers' equation*

  $$\partial_t u + \partial_x \left(\frac{1}{2} u^2 \right) = 0 \tag{3.3}$$

  This is a nonlinear equation with the flux function being convex in u. It is basically the simplest model that includes nonlinear effects of fluid dynamics. In contrast to the linear advection equation, Burgers' equation creates shocks [LeV92].

- *Buckley-Leverett equation*

  $$\partial_t u + \partial_x \left(\frac{u^2}{u^2 + \frac{1}{2}(1-u)^2} \right) = 0 \tag{3.4}$$

  As the most complex case considered here, this nonlinear equation contains a flux function which is non-convex in u. It is a model for two phase fluid flow in a porous medium, as it is applied in oil reservoir simulation [LeV92].

When considering numerical schemes for computing solutions of these equations, one is mainly interested in discontinuities. These discontinuities are either already part of the initial condition or emerge during the evolution because of the appearance of shocks.
Unfortunately, discontinuities are not only the most interesting part of the solution, they also cause most of the problems in the numerical schemes. Basically, one has the choice between schemes that lead to oscillations at discontinuities, or schemes that introduce an artificial viscosity and therefore smear across discontinuities.

A popular strategy to avoid this dilemma is the smart mixture of both types of methods, such that oscillations are prevented while the amount of artificial diffusion is kept as small as possible. Such methods are called *TVD methods* or *high resolution methods* as they both diminish the total variation and yield a high resolution at shocks [LeV92].
The concept to distinguish between smooth areas and shocks in order to choose the mixture coefficients in TVD methods has further led to an even more advanced type of numerical schemes, the so-called *essentially non-oscillatory (ENO) schemes* [HOEC86, Har87, Son97] and *weighted ENO (WENO) schemes* [LOC94]. In the presence of discontinuities, these semi-discrete schemes interpolate with a high order polynomial that is constrained to introduce

as little oscillations as possible. Due to the continuous modelling of discontinuities, these schemes can yield a very high shock resolution. On the other hand, ENO schemes are quite complex, which has negative influence on computational efficiency and amenability for theoretical analysis.

There is another possibility to deal with the above-mentioned problems while keeping things simple. The idea is to decide for one type of scheme, either a second order oscillating one or a first order monotone one, and to remove the typical artifacts by an adequate filtering process. For a second order scheme, which introduces oscillations, an appropriate nonlinear diffusion filter is to be employed, while a shock filter or inverse diffusion filter is the right choice in order to remove the dissipative artifacts of a first order scheme. In the following both strategies are discussed.

3.1 Second Order Lax-Wendroff Improved by TV Flow

A popular second order method is the Richtmyer two-step Lax-Wendroff scheme [LW60, RM94]:

$$\begin{aligned} u_{i+\frac{1}{2}}^{k+1/2} &= \frac{1}{2}(u_i^k + u_{i+1}^k) - \frac{\tau}{2h}\left(f(u_{i+1}^k) - f(u_i^k)\right) \\ u_i^{k+1} &= u_i^k - \frac{\tau}{h}\left(f(u_{i+\frac{1}{2}}^{k+1/2}) - f(u_{i-\frac{1}{2}}^{k+1/2})\right) \end{aligned} \tag{3.5}$$

with time step size τ and spatial grid size h, which is $h = 1$ unless stated otherwise. Fig. 3.1 shows a solution computed with Lax-Wendroff for the linear advection equation together with the initial condition. One can clearly make out the oscillations introduced by the scheme.

From a signal processing point of view, these oscillations can be regarded as noisy artifacts which can be reduced by means of an adequate filter. For this purpose, Engquist et al. [ELS89] designed a special nonlinear filter which removes oscillations but seeks to leave the remainder of the solution untouched.

The works of Grahs et al. [GMS02, GS02] and Wei [Wei02] are based on the same basic concept, however, they employ a nonlinear anisotropic diffusion filter for removing the oscillations. As nonlinear anisotropic diffusion is described by a partial differential equation, this has the advantage that the dissipation is explicitly known. This is in contrast to the filter in [ELS89] which is defined in an algorithmic way.

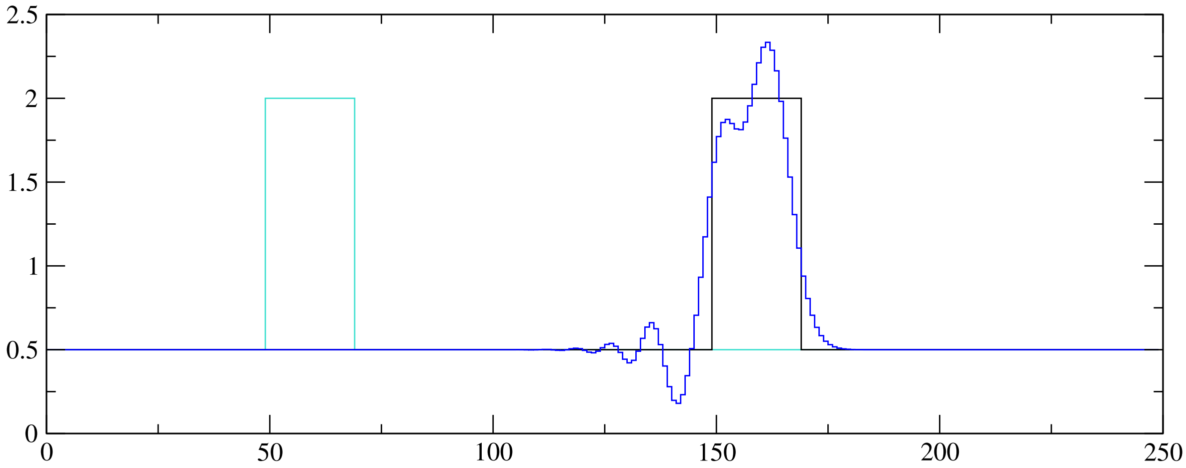


Fig. 3.1: Linear advection. CYAN LINE: Initial condition. BLACK LINE: Correct solution after $t = 100$. BLUE LINE: Lax-Wendroff with $\tau = 0.5$.

Unfortunately, the utilized anisotropic diffusion filters contain lots of parameters, which can be difficult to choose. When applying the numerical scheme in practice, it might be unclear whether a certain effect is due to the physical equation or due to an inappropriate parameter setting.
At this point, the reader may recall TV flow from Section 2.1.1. TV flow is known to remove oscillations very well and it does not contain a contrast parameter like many other types of nonlinear diffusion. Thus it makes sense to consider a TV diffusion filter in order to deal with the oscillations emerging from the Lax-Wendroff method. This has been suggested in [BGS02, Bür05]. In comparison to the filter in [ELS89] and the anisotropic diffusion filter in [GMS02], the method is simpler and therefore better to analyze.

An important issue is the coupling of the Lax-Wendroff scheme and the diffusion process. It can be achieved by a multiplicative splitting, i.e., by alternating the scheme in (3.5) and discretized TV flow[1]:

$$\begin{aligned}
u_{i+1/2}^{k+1/3} &= \frac{1}{2}(u_i^k + u_{i+1}^k) - \frac{\tau}{2h}\left(f(u_{i+1}^k) - f(u_i^k)\right)\\
u_i^{k+2/3} &= u_i^k - \frac{\tau}{h}\left(f(u_{i+1/2}^{k+1/3}) - f(u_{i-1/2}^{k+1/3})\right)\\
u_i^{k+1} &= u_i^{k+2/3} + \tau\alpha\left(\frac{u_{i+1}^{k+2/3} - u_i^{k+2/3}}{\sqrt{\left(u_{i+1}^{k+2/3} - u_i^{k+2/3}\right)^2 + \epsilon^2}} - \frac{u_i^{k+2/3} - u_{i-1}^{k+2/3}}{\sqrt{\left(u_i^{k+2/3} - u_{i-1}^{k+2/3}\right)^2 + \epsilon^2}}\right)
\end{aligned} \tag{3.6}$$

where $\epsilon > 0$ is a small constant for stabilizing the numerics of TV flow as discussed in Section 2.1.3. The coupling parameter α determines the amount of TV flow added to the scheme. In [GMS02] it was pointed out that the amount of artificial diffusion has to be scaled with the spatial grid size h in order to ensure consistency. This scaling exactly cancels out in the scheme when TV flow is involved.
Note that the third step in (3.6) may be replaced by several TV flow iterations with time step size smaller $\tau\alpha$ or by a semi-implicit implementation of TV flow. This is necessary as soon as $\tau\alpha > \frac{\epsilon h^2}{2}$.

The parameter ϵ is only needed to ensure stability of the TV flow implementation. It can be set to a fixed value, keeping in mind that it has to be scaled linearly with the contrast in the data. In the examples shown here, it has been set to $\epsilon = 0.0001$. Using the two-pixel numerics for implementing the third step of (3.6), this parameter disappears completely.
The amount of supplemented TV flow α, however, remains as a free parameter in the scheme. Alternatively, it has been suggested in the scope of TV regularization to vary α locally and to determine it by means of the local variance in the data [BGS02].

Fig. 3.2 shows the impact of TV flow on the Lax-Wendroff scheme. Applying the combined scheme to the linear advection equation, one can see that thanks to TV flow the oscillations are removed whereas the rest of the solution hardly suffers from the filtering operation.

Unfortunately, the concept does not work that well with other settings. Fig. 3.3 displays the performance of the method in case of Burgers' equation. Obviously, a larger amount of TV flow needs to be supplemented here to avoid oscillations in the solution. However, this introduces a significant phase shift. This may be explained by the large amount of mass that

[1] See also Section 2.1.3

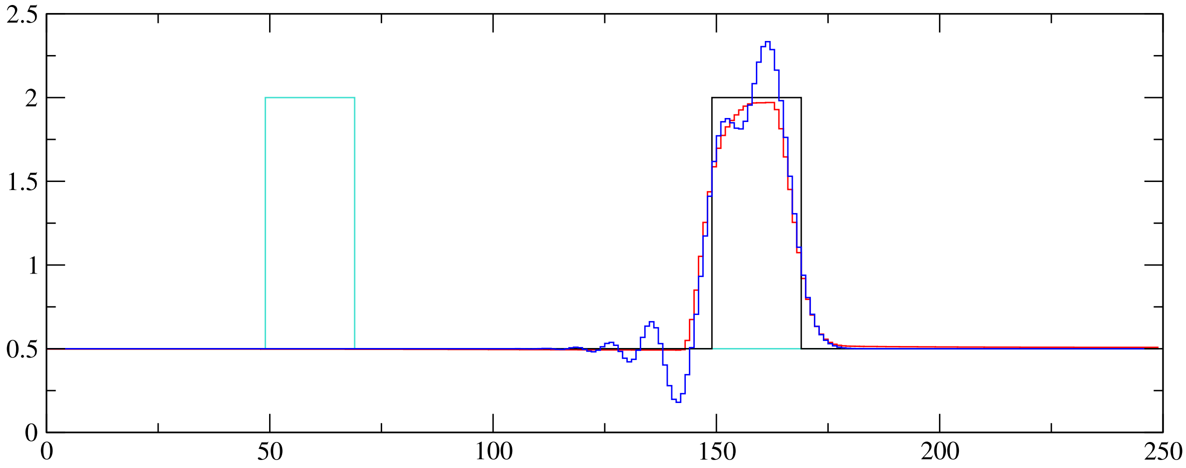


Fig. 3.2: Linear advection. Cyan Line: Initial condition.
Black Line: Correct solution after $t = 100$. Blue Line: Lax-Wendroff with $\tau = 0.5$.
Red Line: Combined Lax-Wendroff and TV flow with $\tau = 0.5$ and $\alpha = 0.01$.

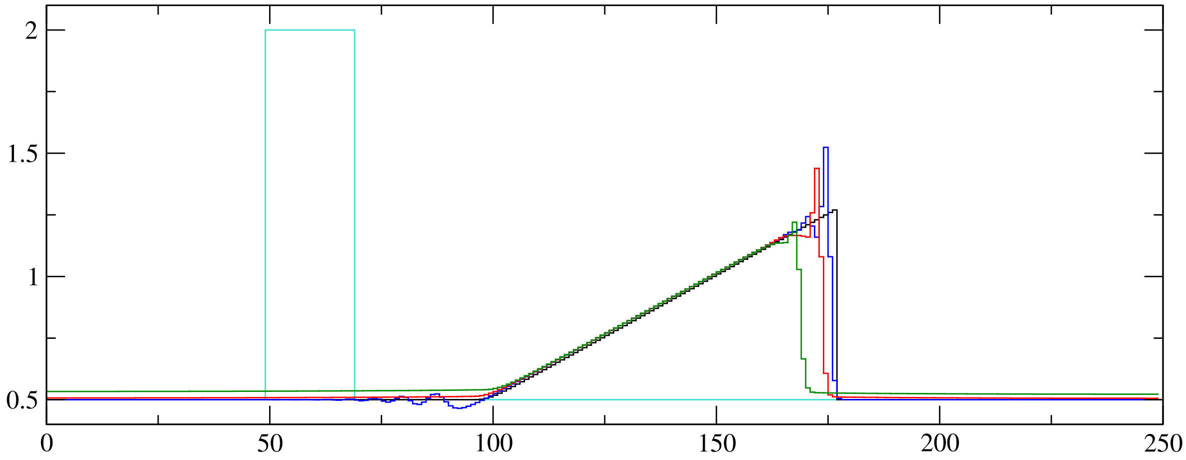


Fig. 3.3: Burgers' equation. Cyan Line: Initial condition.
Black Line: Correct solution after $t = 100$. Blue Line: Lax-Wendroff with $\tau = 0.5$.
Red Line: Combined Lax-Wendroff and TV flow with $\tau = 0.5$ and $\alpha = 0.01$.
Green Line: Combined Lax-Wendroff and TV flow with $\tau = 0.5$ and $\alpha = 0.03$.

is spuriously distributed into the flat regions on the left and the right. Note that the phase error also occurs for $\alpha = 0.01$ where the solution still contains considerable oscillations. Thus the problem cannot be solved by smaller α. It is a constitutional problem of TV flow which considers also the two large boundary regions as oscillation.

Similar problems appear when the scheme is applied to the Buckley-Leverett equation. Also here, TV flow transfers mass from the left boundary region to the right one. Although the shock is closer to its correct position than in the unfiltered version, the result is not satisfactory. Interestingly, replacing TV flow by simple homogenous diffusion $u_t = u_{xx}$ leads to much better results.
A possible way to improve the results is to restrict the filtering to local areas with an unreasonable entropy production [GS02]. This resembles the concept in [BGS02] to change α locally by means of the local variance. More details about numerical schemes that apply TV filtering in order to remove oscillation artifacts in hyperbolic conservation laws can be found in the thesis of Bürgel [Bür05].

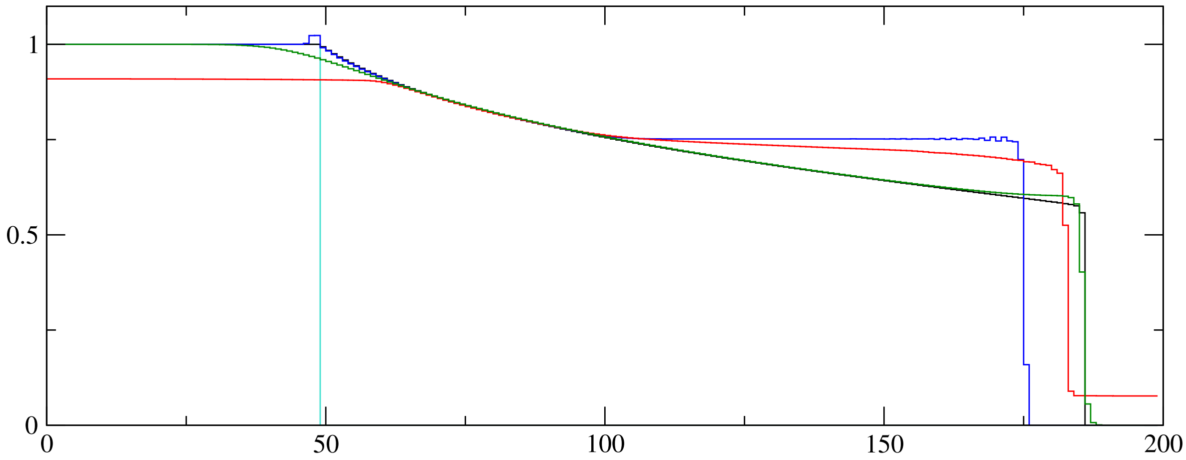


Fig. 3.4: Buckley-Leverett equation. Cyan Line: Initial condition.
Black Line: Correct solution after $t = 100$. Blue Line: Lax-Wendroff with $\tau = 0.5$.
Red Line: Combined Lax-Wendroff and TV flow with $\tau = 0.5$ and $\alpha = 0.05$.
Green Line: Combined Lax-Wendroff and homogeneous diffusion with $\tau = 0.5$ and $\alpha = \frac{0.25}{h}$.

3.2 First Order Upwind Improved by Nonlinear Inverse Diffusion

The approach of the last section has been based on the removal of oscillation artifacts introduced by a second order scheme. On the other hand, there are also monotone first order schemes for hyperbolic conservation laws that do *not* produce oscillations at shocks. In return, they suffer from strong dissipation, since they involve a significant amount of numerical diffusion to achieve their favorable stability properties.

Taking such a scheme as starting point, the goal of an adequate filtering technique is not to remove oscillations while keeping the shock resolution, but, vice-versa, to improve the shock resolution while keeping monotonicity. Ideally, such an improved monotone scheme should also turn the first order basis method having low accuracy into a higher accurate second-order method. This can be achieved by applying an appropriate filter not only at shocks but also in other areas of the solution in order to balance the immoderate numerical viscosity of the first order scheme.

The basic idea of such a procedure is not completely new, but is the main concept of the class of flux-corrected transport (FCT) schemes [BB73, BBH75, BB76, Zal79]. The scheme introduced in this chapter, however, offers the advantage of being applicable also to the important class of *nonlinear* conservation laws. This is in contrast to classical FCT schemes which are originally restricted to linear processes.

Upwind scheme. Starting point of the suggested method is the Upwind scheme:

$$u_i^{k+1} = u_i^k - \frac{\tau}{h}\left(f(u_i^k) - f(u_{i-1}^k)\right). \qquad (3.7)$$

If the usual CFL condition is satisfied [CFL28, LeV92], this scheme is a generalized monotone scheme, i.e., it is local extremum diminishing (LED). Furthermore, it does not introduce new extrema during a computation, i.e., it diminishes the number of extrema (NED).

However, the Upwind scheme has a severe disadvantage: it suffers from undesirable blurring effects, see Fig. 3.5. To quantify these viscous artifacts the scheme (3.7) can be written in its *viscous form*

$$\underbrace{u_i^{k+1} = u_i^k - \frac{\tau}{2h}\left(f(u_{i+1}^k) - f(u_{i-1}^k)\right)}_{\text{(A)}} + \underbrace{\frac{\tau}{h}Q_i^{+,k}\left(u_{i+1}^k - u_i^k\right) - \frac{\tau}{h}Q_i^{-,k}\left(u_i^k - u_{i-1}^k\right)}_{\text{(B)}}. \tag{3.8}$$

The underlying idea behind this decomposition is to consider part (A) as a second order approximation of (3.1) in space (and first order in time), while part (B) is (in leading order) the discrete counterpart of the numerical diffusion incorporated in the method.

One easily verifies that (3.7) and (3.8) can be made identical by choosing viscosity coefficients Q_i^+ and Q_i^- that satisfy

$$Q_i^{+,k} = \frac{f(u_{i+1}^k) - f(u_i^k)}{2(u_{i+1}^k - u_i^k)} \quad \text{and} \quad Q_i^{-,k} = \frac{f(u_i^k) - f(u_{i-1}^k)}{2(u_i^k - u_{i-1}^k)}. \tag{3.9}$$

for $u_{l+1}^k \neq u_l^k$, $l \in \{i,\, i{-}1\}$. Note that the assumption $f'(u) \geq 0$ ensures that the viscosities $Q_i^\pm$ are nonnegative. Since the viscosities are proportional to the diffusion coefficients, it follows that forward diffusion takes place. This numerical diffusion is responsible for the undesirable blurring effects that emerge with this first order method. One observes that, in spite of the simplicity of the Upwind scheme, an inherent diffusion process with nonlinear viscosities $Q_i^\pm$ is involved. These nonlinear viscosities are inversely proportional to the derivative of u. In this respect, they closely resemble the diffusivities of TV flow.

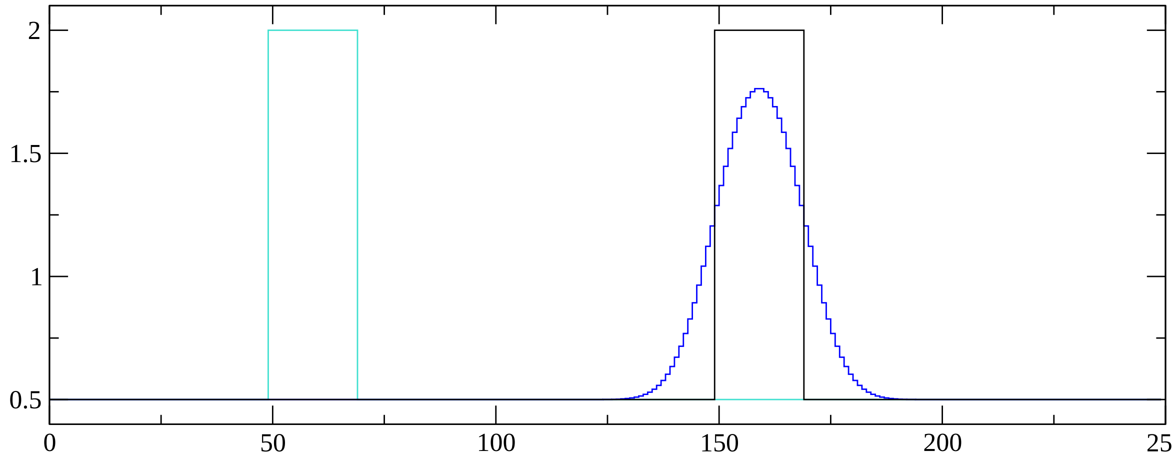


Fig. 3.5: Linear advection. Cyan Line: Initial condition.
Black Line: Correct solution after $t = 100$. Blue Line: Upwind scheme with $\tau = 0.5$.
The numerical diffusion of the Upwind scheme creates severe blurring artifacts.

Stabilized nonlinear inverse diffusion. Since with (3.9) the numerical diffusion coefficients are known, it becomes possible to improve the Upwind scheme by applying a filter that negates exactly this amount of diffusion. Let $u_i^{k+1/2}$ denote the data obtained with the Upwind scheme (3.7) from u_i^k. Then the inverse diffusion step that removes the numerical diffusion reads:

$$u_i^{k+1} = u_i^{k+1/2} - \frac{\tau}{h}\left(Q_i^{+,k+1/2}\left(u_{i+1}^{k+1/2} - u_i^{k+1/2}\right) - Q_i^{-,k+1/2}\left(u_i^{k+1/2} - u_{i-1}^{k+1/2}\right)\right). \tag{3.10}$$

One may argue that the complete removal of the viscous part (B) in (3.8) leads to the instable scheme (A). This is indeed true. To keep the stability properties of the Upwind scheme, it is therefore necessary to stabilize the inverse diffusion process in such a way that it does not

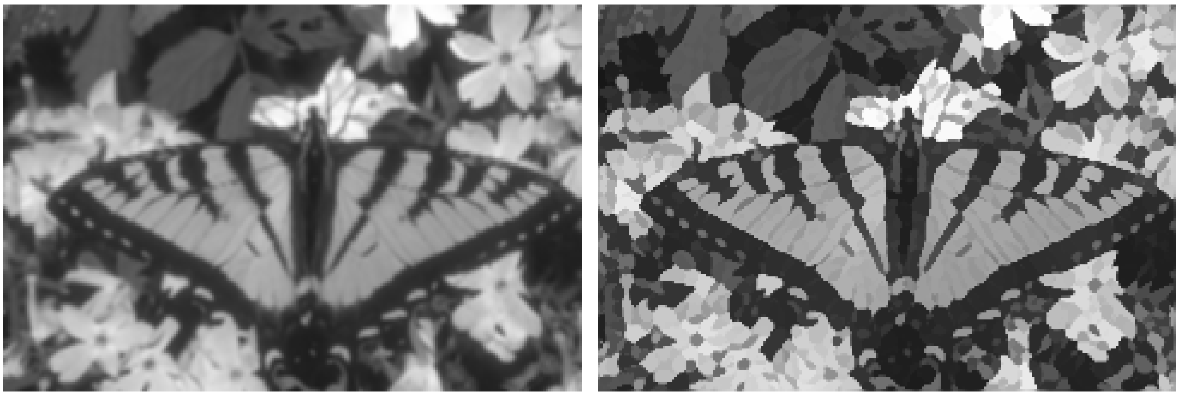


Fig. 3.6: LEFT: Original image with blurred edges.
RIGHT: Image sharpened with stabilized inverse diffusion [OR91].

create new extrema in the solution.

The concept of stabilized inverse diffusion is known from image enhancement, where it has been utilized for the purpose of edge sharpening [OR91, GSZ02]. A very popular alternative in image enhancement are variational deconvolution techniques [YK96].

In [GSZ02] stabilization of inverse diffusion is achieved by coupling it with nonlinear forward diffusion. The resulting filter applies inverse diffusion only for a limited range of slopes, while for all other slopes, forward diffusion ensures stability and the removal of image noise. Designed for image enhancement, this kind of stabilization is not well applicable for the inverse diffusion in (3.10).

Variational deconvolution performs a similar task as inverse diffusion by minimizing the energy functional [YK96]

$$E(u) = \int_{\mathbb{R}} \left((K * u - I)^2 + \alpha |u_x|^2 \right) \, dx \tag{3.11}$$

with the blurring kernel K and the input signal I. Thus one searches a solution u that, when it is convolved with the kernel K, leads to the observed signal I. The second term of the functional is a regularizer that ensures a smooth solution. Unfortunately, the gradient descent that is usually employed for minimizing (3.11) converges very slowly. Moreover, the solution in general contains oscillations. Albeit these oscillations are limited in their magnitude, applying such a method to hyperbolic conservation laws does not yield a monotone scheme. It was tried in the course of this thesis to apply a variational deconvolution technique for reverting the numerical diffusion of the Upwind scheme, yet the results have not been satisfactory.

A much more successful approach is based on the filter proposed in [OR91]. This filter stabilizes the space-discrete version of the inverse heat equation

$$\partial_t u = -u_{xx}. \tag{3.12}$$

Its concept is to stop the inverse diffusion at extrema, thereby explicitly inhibiting over- and undershoots. A continuous model for this stabilized process can be written as:

$$\partial_t u = (g(|u_x|)u_x)_x \qquad \text{with } \; g(|u_x|) = \begin{cases} -1 & \text{if } |u_x| \neq 0 \\ 0 & \text{else} \end{cases} \tag{3.13}$$

yet as the model itself creates shocks, u_x does not exist at the shock location, thus (3.13) is very difficult to analyze. However, we are anyway interested in the space-discrete setting.

Starting with a discretization of (3.12):

$$\begin{aligned} u_i^{k+1} &= u_i^k - \tfrac{\tau}{h}\left(g_{i+1/2} - g_{i-1/2}\right) \\ g_{i+1/2} &= u_{i+1}^k - u_i^k \end{aligned} \tag{3.14}$$

the filter in [OR91] applies the *minmod* function

$$\mathrm{minmod(a, b, c) := sgn(b) \max\big(0, \min(sgn(b)a, |b|, sgn(b)c)\big)} \tag{3.15}$$

in order to obtain new, stabilized flux coefficients

$$\tilde{g}_{i+1/2} = \mathrm{minmod}\left(g_{i-1/2}, g_{i+1/2}, g_{i+3/2}\right) \tag{3.16}$$

which are 0 in the presence of extrema. One can verify that the resulting flux limited inverse diffusion process is ensured not to yield over- or undershoots. Indeed it has been shown in [OR91] that the filter satisfies a local minimum-maximum principle. Fig. 3.6 depicts the corresponding 2-D filter applied to a blurred image. Obviously, edges are enhanced considerably, whereas the filter does not introduce any oscillations. Furthermore, the filter is mass conservative, which is very important for its application in hyperbolic conservation laws.

This type of stabilization appears appealing to design an inverse diffusion filter that negates the numerical diffusion of the Upwind scheme. Whereas the inverse diffusion filter in [OR91] negates *linear* diffusion, however, the Upwind scheme necessitates a filter that can deal with the nonlinear diffusion coefficients emerging from the viscous form in (3.8).

Fortunately, an extension to the nonlinear case turns out to be straightforward. Setting the flux coefficients to

$$g_{i+1/2} = Q_i^{+,k+1/2}\left(u_{i+1}^{k+1/2} - u_i^{k+1/2}\right) \tag{3.17}$$

the minmod function can be employed to obtain

$$\tilde{g}_{i+1/2} = \mathrm{minmod}\left(g_{i-1/2}, g_{i+1/2}, g_{i+3/2}\right). \tag{3.18}$$

This yields the following stabilized version of the filter in (3.10):

$$u_i^{k+1} = u_i^{k+1/2} - \frac{\tau}{h}\left(\tilde{g}_{i+1/2} - \tilde{g}_{i-1/2}\right). \tag{3.19}$$

The resulting numerical scheme that alternates the Upwind scheme and inverse diffusion according to (3.19) has some very pleasant properties:

- It satisfies a local discrete minimum-maximum principle. This follows from the fact that both the Upwind scheme and the inverse diffusion step satisfy such an extremum principle. More details can be found in [BBSW05].
- In contrast to contemporary high resolution methods, the scheme is simple and hence better accessible to a profound theoretical analysis. Furthermore, and in contrast to the approach in Section 3.1, there are no tuning parameters in the filtering step, since the necessary amount of inverse diffusion can be derived directly from the underlying Upwind scheme.

Fig. 3.6 further shows that the basic stabilized inverse diffusion scheme from [OR91] is well applicable to two-dimensional data. Thus it is expected that the scheme can be generalized quite easily to higher dimensions.

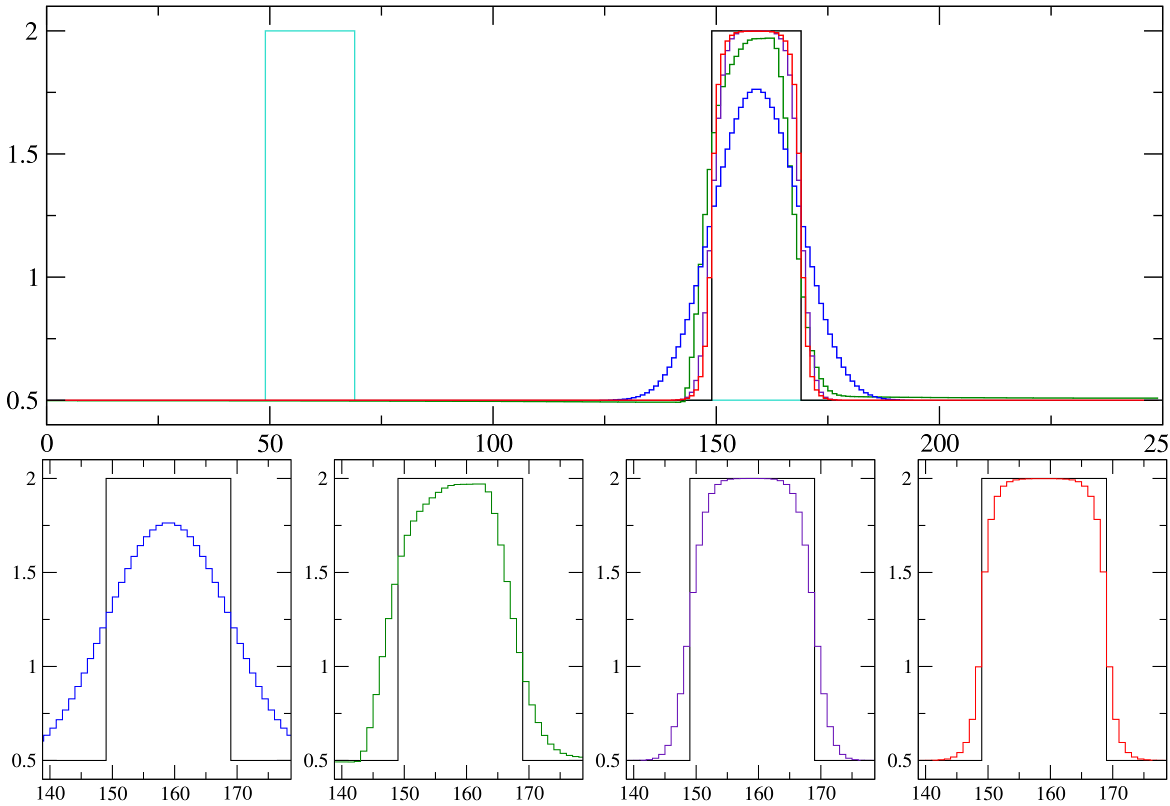


Fig. 3.7: Linear advection with $t = 100$ and $\tau = 0.5$. BOTTOM: ZOOM.
CYAN LINE: Initial condition. BLACK LINE: Correct solution. BLUE LINE: Conventional Upwind.
GREEN LINE: Lax-Wendroff filtered with TV flow ($\alpha = 0.01$). INDIGO LINE: TVD method.
RED LINE: Upwind with inverse diffusion.

Experiments. Like in the last section, the method is evaluated in three different test cases, which comprise linear advection, the square-wave solution of Burgers' equation, and the numerical solution of a Riemann problem for the Buckley-Leverett equation. Additionally, an order test determines the empirical order of convergence of the new scheme.

In the experiments, the scheme is also compared to a contemporary TVD method with the van Leer slope limiter. For details on this method see [GR91, LeV92]. Note that there is a wide variety of possibilities to obtain higher order accuracy in standard TVD schemes, e.g., flux limiting, slope limiting, or ENO schemes. The chosen TVD method is supposed to be a good representative for this class of techniques and the used test cases.

Fig. 3.7 - Fig. 3.9 show that the new scheme (red line) compares favorably to alternative schemes. In particular, one can see the improved accuracy compared to the Upwind scheme without an inverse diffusion step (blue line). Also its comparison to the combination of an oscillating second order method with TV diffusion (green line) indicates that the strategy to start with a monotone scheme may have advantages compared to the approach presented in the last section. Particularly note that the new scheme does not involve any tuning parameters as the amount of inverse diffusion is given explicitly by the Upwind scheme.

The scheme can even keep up with the performance of a TVD scheme. Though the results are not the same, the absolute total error is approximately the same.

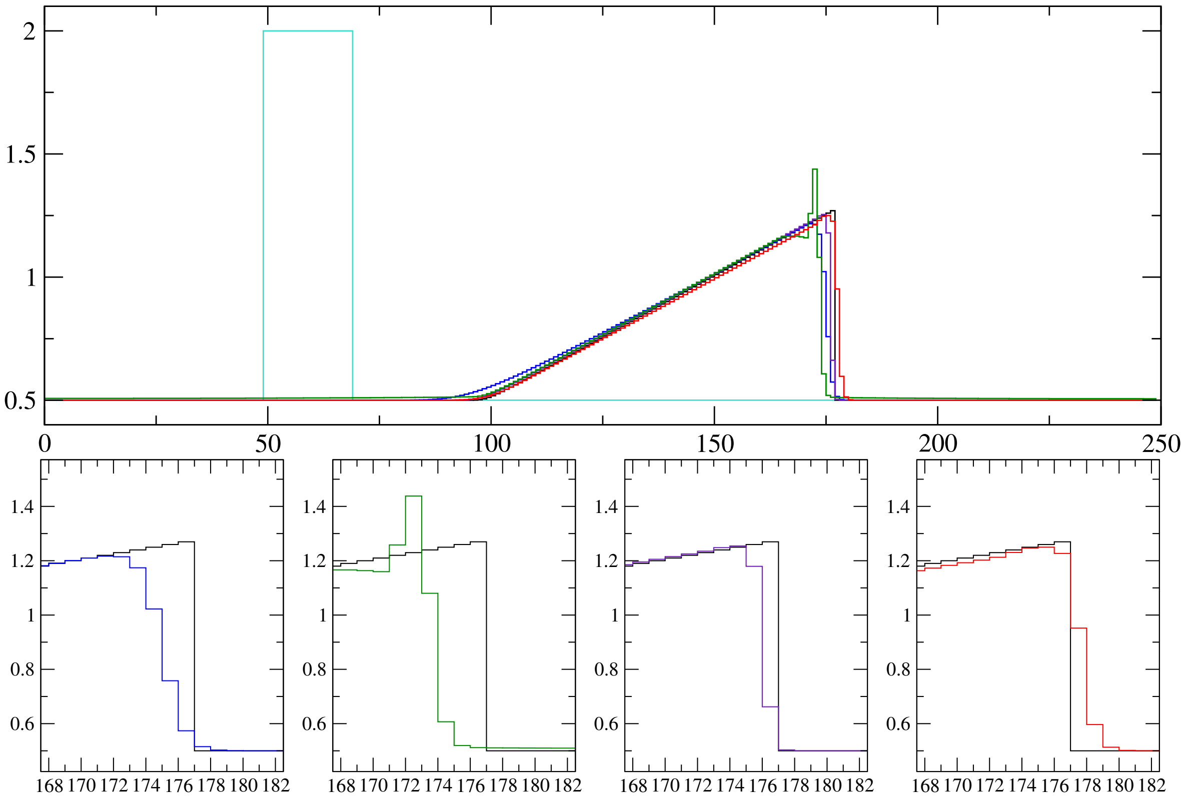


Fig. 3.8: Burgers' equation with $t = 100$ and $\tau = 0.5$. Bottom: Zoom.
Cyan Line: Initial condition. Black Line: Correct solution. Blue Line: Conventional Upwind.
Green Line: Lax-Wendroff filtered with TV flow ($\alpha = 0.01$). Indigo Line: TVD method.
Red Line: Upwind with inverse diffusion.

♯ nodes	h	τ	time steps	L_1-error	EOC
20	0.1	0.0001	20000	0.394969	-
40	0.05	0.0001	20000	0.135555	1.54286
80	0.025	0.0001	20000	0.0508049	1.41584
160	0.0125	0.0001	20000	0.0147794	1.78138
320	0.00625	0.0001	20000	0.00460051	1.68372

Tab. 3.1: Arrangement of the computational parameters for the numerical convergence study together with the corresponding L_1-error and the experimental order of convergence (EOC).

This qualitative behavior is affirmed by the order test[2] shown in Tab. 3.1. For this test, the smooth initial data

$$u(x, 0) = \sin(\pi x)$$

has been propagated by the linear advection equation on a grid over $[-1, 1]$ with periodic boundary conditions. Thus after time $t = 2$ the correct solution of the differential equation exactly matches the initial condition. By measuring the error $e(h)$ of the numerical solution in the L_1-Norm for a sequence of spatial grids with diminishing mesh widths h, one can determine the *experimental order of convergence* EOC defined by

$$\text{EOC} := \frac{\log\left(\frac{e(h)}{e(h/2)}\right)}{\log(2)}.$$

[2] which has been kindly provided by Michael Breuß

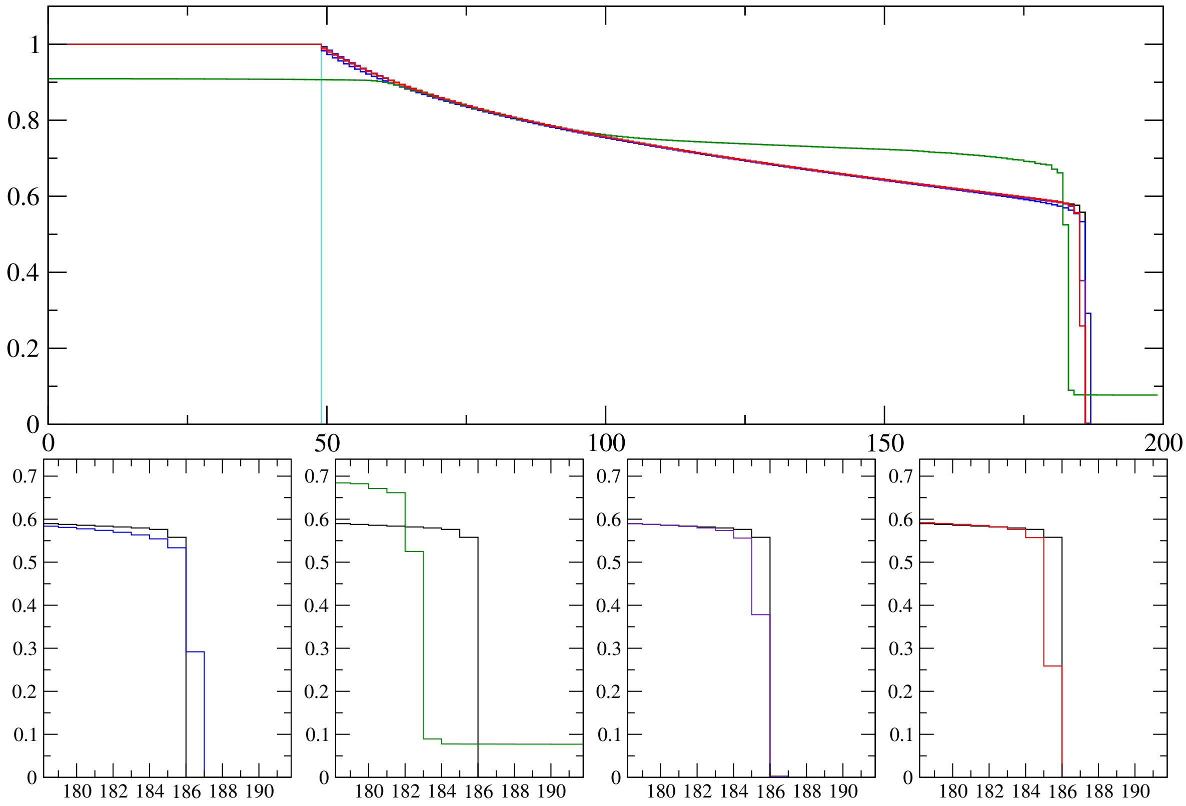


Fig. 3.9: Buckley-Leverett equation with $t = 100$ and $\tau = 0.5$. Bottom: Zoom.
Cyan Line: Initial condition. Black Line: Correct solution. Blue Line: Conventional Upwind.
Green Line: Lax-Wendroff filtered with TV flow ($\alpha = 0.05$). Indigo Line: TVD method.
Red Line: Upwind with inverse diffusion.

The results show that the inverse diffusion turns the classical Upwind scheme from first order to nearly second order.

The most interesting issues that are subject to future research are the application of the scheme to vector-valued conservation laws and its extension to higher dimensions. Actually this extension is quite straightforward, since both the Upwind scheme and the inverse diffusion filter are available also for higher dimensions. However, the inverse diffusion filter proposed in [OR91] is not rotationally invariant, i.e., transporting a circular structure will lead to a rectangular solution. Finding a remedy to this problem is not easy, especially since the filter still has to interact with the Upwind scheme and its numerical diffusion.

4 Low Level Vision

We've all heard that a million monkeys banging on a million typewriters
will eventually reproduce the entire works of Shakespeare.
Now, thanks to the Internet, we know that this is not true.

Robert Wilensky
Academic

Chapter 2 has dealt with PDE-based techniques for signal processing. Although the focus has been put on images, the techniques presented there are general enough to process also other data. Such an example has been shown in Chapter 3.

In this chapter we will turn back to images with the motivation to extract basic features that play an important role in visual perception and are thus useful for image segmentation in Chapter 5. This task is called here *low level vision* in order to indicate the basic stage of the computer vision task: the processed structures still have the form of images and the data is not yet interpreted in any of the methods.

Two main topics are examined in this chapter: texture and optic flow. The term *texture* is unfortunately a quite loosely defined expression. Although it is clear what is meant with texture in the broad sense, there is no general compliance on which task a texture model has to satisfy. Which parts of an image should be modelled as texture and which parts as separate objects? In this thesis, texture is regarded as local, small scale structure and motivated as input feature for image segmentation. This restriction yields the possibility to develop a feature model that works well together with image segmentation. It should be noticed that texture may also be seen in a much broader sense, yet this is beyond the scope of this thesis.

In contrast to texture, the term *optic flow* is very well defined: it is the displacement field between two images, and it is feasible to work out a sound variational model to estimate this displacement field. Optic flow is mostly used in motion analysis, yet it is also applicable in the scope of image registration or stereo vision. Apart from that, optic flow can serve as cue for structure from motion, image segmentation, and tracking. Thus, like texture, also optic flow is used as feature for image segmentation in Chapter 5.
Note that optic flow also includes the computation of disparity from a pair of stereo images. Disparity would be a further useful feature for image segmentation. However, optic flow is computed here without respecting the so-called *epipolar constraint* that emerges from a calibrated camera system, so one solves a much harder problem than actually necessary. As this has negative effects on the results when compared to methods specifically designed for computing disparity, the possibility to handle stereo images is only briefly discussed in an experiment at the end of this chapter. A technique that computes the disparity including the epipolar constraint, and which is similar to the optic flow technique suggested in this chapter, can be found in [RD96, ADSW02].

The general idea in this chapter - in particular when considering the case of optic flow computation - is the extension of the sound mathematical modelling from Chapter 2 to more complex problems than image denoising, showing that theoretical foundation also goes well together with successful algorithms of higher degree of complexity. Large parts of the work on optic flow estimation have been in cooperation with Andrés Bruhn and Nils Papenberg.

4.1 Texture Discrimination

As already mentioned, the exact definition of texture is not easy. In the early work of Julesz [Jul81, Jul86], texture has been defined as the repetition of basic image elements, so-called *textons*. From the theoretical point of view, textons are quite attractive. They can be assembled by so-called *sketches*, which model the grouping of the textons. A recent theoretical overview based on this concept has been presented in [ZGWW02].

The arising question is how to extract textons from the image. Assuming that a texture can be represented by a linear combination of basis functions, one can measure at every position how much each basis function correlates with the image. This comes down to a convolution of the image with the basis functions. Using these convolution results as weights in the linear combination, the texture can be regenerated approximatively. By modelling also the stochastic process behind the grouping of basis functions, it becomes even possible to create new, similar patches of the texture. This is often done by means of Markov models, see e.g. [WZG02].

The problem of this extraction procedure is the large number of basis functions that is necessary to represent a texture properly. At this point, it should be noted that a complex texture representation which can regenerate a texture very well need not necessarily be suitable for texture discrimination. As long as two different textures are not separated yet, i.e. before segmentation, it is very difficult to estimate simultaneously both the complex texture models and the boundary between the textures.

For this reason, often simplified *discriminative texture models* are used for segmentation purposes. In these models the number of basis functions is greatly reduced. Or it is assumed that textures can be distinguished by means of certain features extracted in a local neighborhood. A variety of such features can be found, e.g, in [RdB93]. The advantage of discriminative texture models is that they lead to a low-dimensional feature space. Hence, though they cannot describe textures in full detail, they are well suited for pre-attentive vision tasks, where a segmentation is not available yet, and the texture is to serve as cue in order to facilitate segmentation.

Gabor filter bank. The most popular set of basis functions is described by a so-called *Gabor filter bank* [Gab46]. Each basis function in a Gabor filter bank has one preferred orientation and scale. Thus convolution of the image with each filter yields high responses at those positions in the image where the local structure fits the orientation and scale of the filter.

Gabor filters have attained high popularity, in particular because neuroscience has found a similar behavior of neuronal cells in the primary visual cortex of primates. Since then, a lot of research aimed on determining a model that best fits the responses measured in the visual cortex [Mar80, Dau85, Fie87]. Moreover, several works on texture segmentation based on Gabor filters are available in the literature, e.g. [BCG90, DHW94, SSZ01, PD02b].

In principle, the Gabor function is a Gaussian function modulated by an oriented sinusoidal wave with orientation ϕ and frequency f:

$$G_{\phi,f}(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \cdot e^{2\pi f i(x\cos\phi + y\sin\phi)}. \tag{4.1}$$

The standard deviation σ of the Gaussian is mostly set dependent on the frequency f of the wave function. This is reasonable, since both parameters of the Gabor function determine the scale of the local structure that is sought to respond to the filter. The relation between σ and f is expressed as [PNNT96]:

$$\sigma = 3\frac{\sqrt{2\ln 2}}{2\pi} f. \tag{4.2}$$

Typically, 4 different orientations and 3 different scales are used for a Gabor filter bank. They are set here to $\phi \in \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$ and $f \in \{0.2, 0.35, 0.5\}$. Fig. 4.1 shows the corresponding filter stencils. Sometimes more filters are applied, but this increases not only the accuracy of sampling but also the redundancy of the filter bank and the dimension of the feature space.

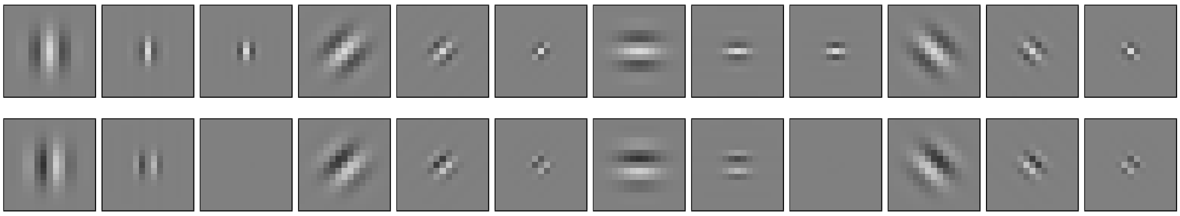


Fig. 4.1: Gabor filter stencils with 4 different orientations $\phi \in \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$ and 3 different frequencies $f \in \{0.2, 0.35, 0.5\}$.
Upper Row: Real part. Lower Row: Imaginary part.

Instead of using directly the responses of the Gabor filters for texture discrimination, mostly the so-called Gabor energy

$$E_{\phi,f} = \sqrt{(I * \Re(G_{\phi,f}))^2 + (I * \Im(G_{\phi,f}))^2} \tag{4.3}$$

is used, where $\Re$ and $\Im$ denote the real and imaginary part, respectively. This reduces the dimension of the feature vector by one half. The Gabor energies for the sample image in Fig. 4.2 are depicted in Fig. 4.3.

Motivation for a sparse texture feature space. Basically the Gabor filter bank extracts nothing else than the magnitude, orientation, and scale of local texture elements. However, these texture properties are hidden in highly redundant features. While this representation is positive for regenerating the texture, for texture discrimination it would be favorable to reduce the redundancy in order to obtain a low-dimensional feature space.

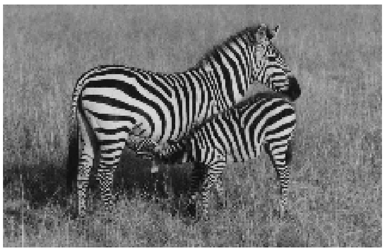


Fig. 4.2: Zebra image (220 × 140).

Following this argumentation, I now show a different way how to extract the texture magnitude, orientation, and scale from the image by using the structure tensor from Section 2.2 and a new measure for scale. This leads to a 4-dimensional feature space holding basically the same information as the 12-dimensional Gabor feature space, what makes segmentation more efficient.

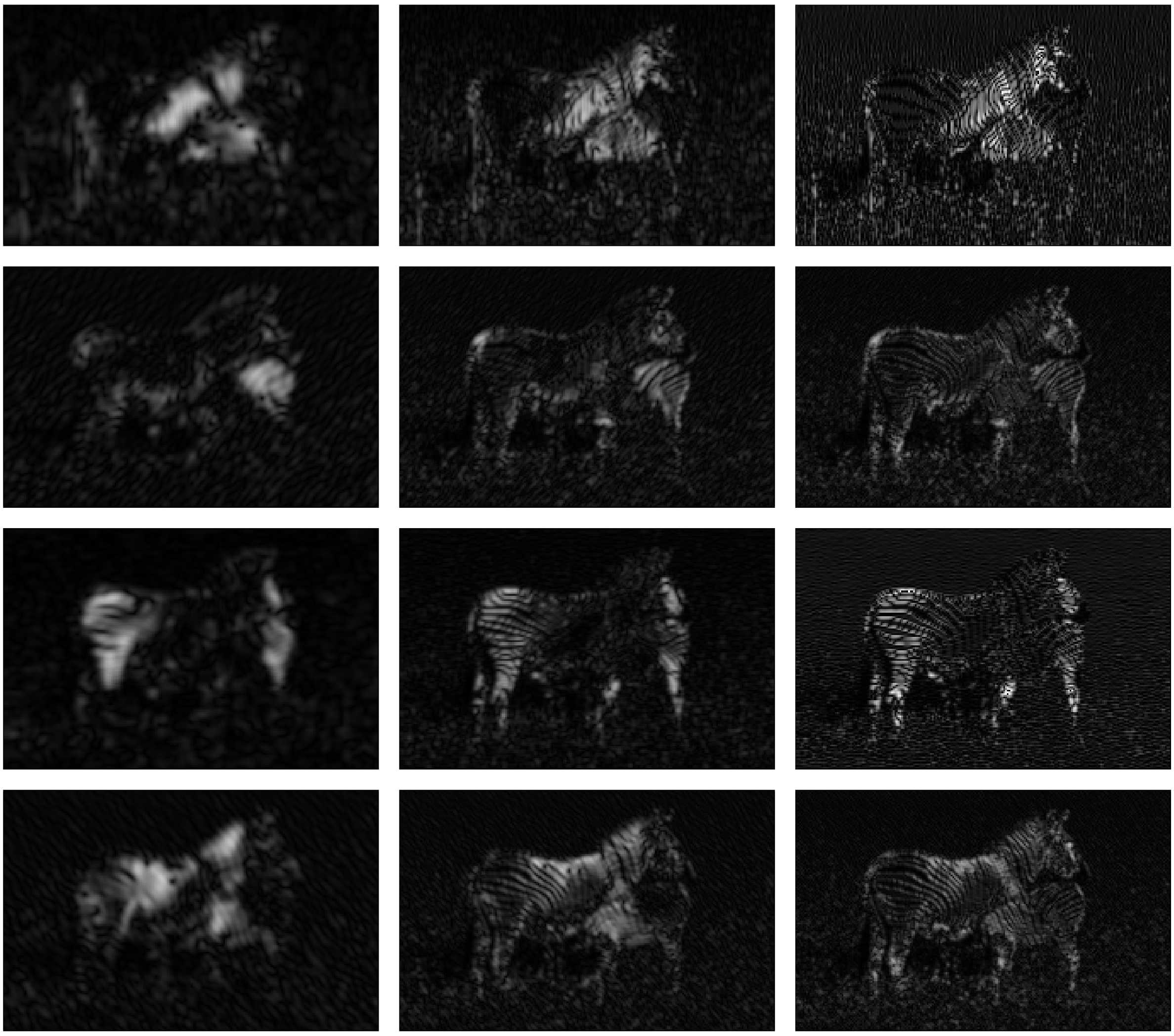


Fig. 4.3: Gabor energy for the filter stencils shown in Fig. 4.1 and the zebra image in Fig. 4.2. Same order as in Fig. 4.1 from left to right, top to bottom.

4.1.1 Texture Magnitude, Orientation, and Homogeneity of Orientation

Already in [RS91] and [BGW91] the structure tensor from Section 2.2 has been suggested for texture discrimination. This is reasonable, as the structure tensor has been introduced as a tool for estimating the orientation of local structure, which is an important feature for discriminating textures. Additionally, the structure tensor contains the magnitude of the structure and, thanks to local smoothing, cues about the homogeneity of the local orientation. All these are reasonable features for texture discrimination.

In Section 2.2.2 it has been stressed that the conventional structure tensor with its Gaussian smoothing involved is not perfect regarding the estimation accuracy. This is because the Gaussian convolution leads to blurring effects. An extension has been introduced, the nonlinear structure tensor, which replaces Gaussian convolution by nonlinear diffusion and therefore avoids the blurring effects. In texture segmentation, this gain in accuracy is very important in order to obtain good discrimination results also near texture boundaries. Fig. 4.4 depicts the components of the nonlinear structure tensor with $p = 1$ for the zebra image in Fig. 4.2 and different amounts of smoothing.

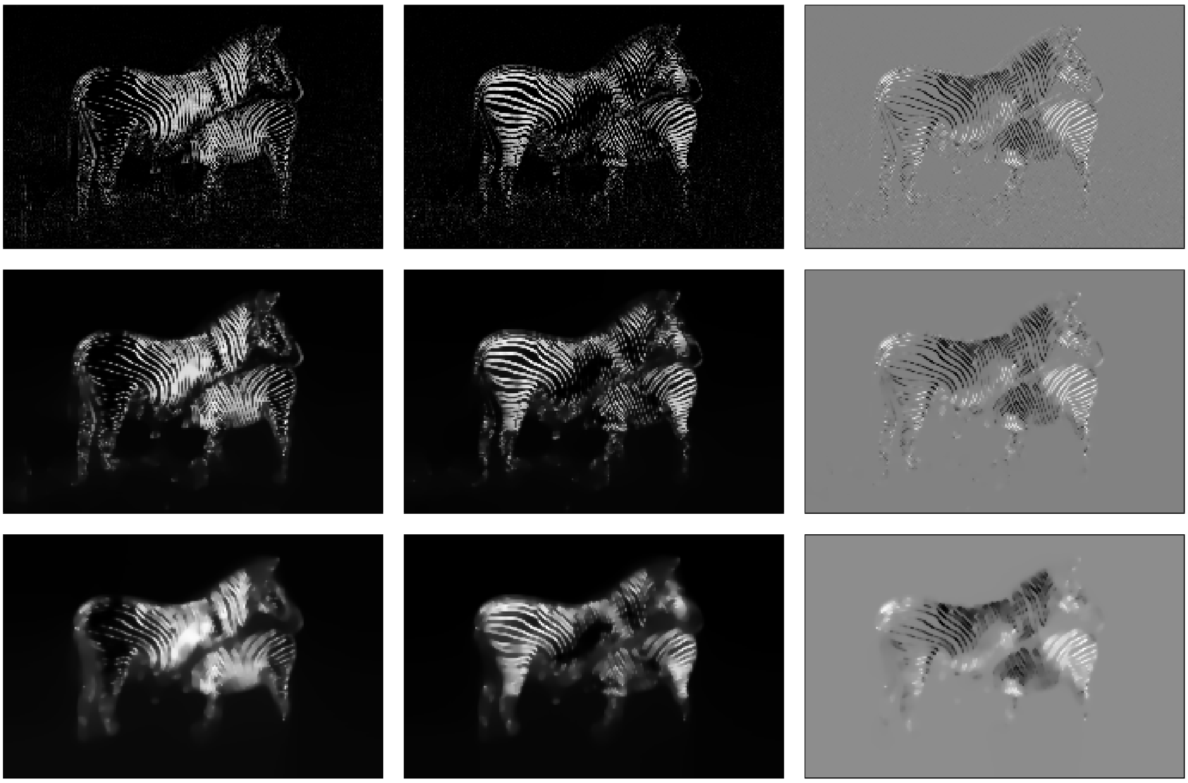


Fig. 4.4: Structure tensor components J_{11}, J_{22}, and J_{12} for the image in Fig. 4.2 and $p = 1$ (TV flow). Top Row: $t = 0$. Middle Row: $t = 2000$. Bottom Row: $t = 5000$.

4.1.2 Texture Scale

Comparing the information contained in the structure tensor with that of Gabor responses, it becomes obvious that an important cue is missing: the scale of texture elements. Like orientation, scale is a very important feature to distinguish textures, thus a measure of local scale is needed to complete the sparse feature space.

The simplest idea for a local scale measure is to consider the variance in a fixed local window. However, such a measure has several drawbacks: large scales with high gradients result in the same value as small scales with low gradients, i.e., local variance is not equivalent to local scale. Moreover, the use of non-adaptive local windows always blurs the data. The latter drawback also appears with the general idea of local Lyapunov functionals [SCT99], which includes the case of local variance. Other works on local scale can be found in [JK92, Lin94, Lin99, EZ98, GMS00].

They all have in common that they are gradient based, i.e., their measure of local scale depends directly on local gradients or their derivatives. Consequently, the scale cannot be measured in regions without a significant gradient. We know this issue already from orientation estimation with the structure tensor. In that case the problem has been addressed by estimating the dominant orientation in a local neighborhood. The same way one could try to estimate the dominant scale in a local neighborhood.

However, we embark here on another strategy that is not edge based but region based. This means, our local scale measure does not depend on the behavior of the gradient in scale space, but directly on the size of regions. This has many advantages considering segmentation purposes, since all state-of-the-art segmentation techniques are based on regions. Moreover, a region based technique does not need the definition of a local window and therefore yields the maximum localization accuracy.

Scale measure based on region-merging. While the motivation for a region based scale measure is clear, it poses the question of how to define regions and how to measure their size. Two different techniques are examined here. First a fast region merging algorithm is considered [KLM94] to obtain a region segmentation according to the cartoon limit of the Mumford-Shah functional [MS85, MS89]

$$E(u,\Gamma) = \int_\Omega (I-u)^2 \, \mathbf{dx} + \alpha \int_\Gamma ds. \tag{4.4}$$

This energy functional searches for a piecewise constant solution u with edges Γ such that u is close to the image and the total edge length is minimal. The parameter α determines to which extend region boundaries are penalized, so larger α yield results with less regions.
The exact optimization of this functional is very difficult. The region merging algorithm therefore uses a greedy heuristic starting with the trivial segmentation where each pixel is a region, and then successively merges those two neighboring regions that lead to the largest energy decrease. This merging of regions yields successively simplified images with increasingly large regions. The size of these regions is the sought local scale $m(\alpha)$.

On the first glance it is surprising that there appears a scale parameter α in the scale measure. On the other hand it is obvious that a pixel lives on several different scales during the simplification of the image by the region merging algorithm. At the beginning, the pixel might be a single noise pixel that has the smallest possible scale until it is merged with other pixels to a small region. This region will further be merged with other regions to form a region of larger scale, and so on.
Certainly one could measure the scale of the regions in the original image, i.e., setting $\alpha \to 0$. However, in real-world images it is very unlikely that two neighboring pixels have exactly the same value, so basically all regions would have size 1. If α is set to a higher value, pixels and regions are merged even if they do not have exactly the same value. Thus by choosing α it is determined how much two regions may differ in their value to be still considered as one region.

A strategy to robustify the scale measurement is to choose not a single value α, but to integrate the results for different α and therefore measuring the *average* scale $\bar{m}$ of the pixel.

$$\bar{m} = \frac{\int_0^{\alpha_{Stop}} m(\alpha)\, d\alpha}{\alpha_{Stop}} \tag{4.5}$$

One might even suggest to compute the average scale up to the α_{Stop} when all region have merged to a single region. However, one is interested in measuring the scale of small scale texture elements. Hence the large scale regions appearing for large α would spoil the measurement. In this spirit, choosing α_{Stop} that large is as counterproductive as choosing $\alpha_{Stop} = 0$: in both cases the scale of regions created by the region merging algorithm does not fit the scale of typical texture elements.

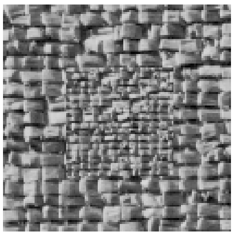


Fig. 4.5: Texture image (123×119).

It can be further useful to use instead of the scale m the inverse scale $\frac{1}{m}$ for the discrimination of textures. Firstly, this restricts the values to a range between 0 and 1, what can become very appealing when the scale measure is combined with other features. Secondly, small scales get a larger contrast relative to large

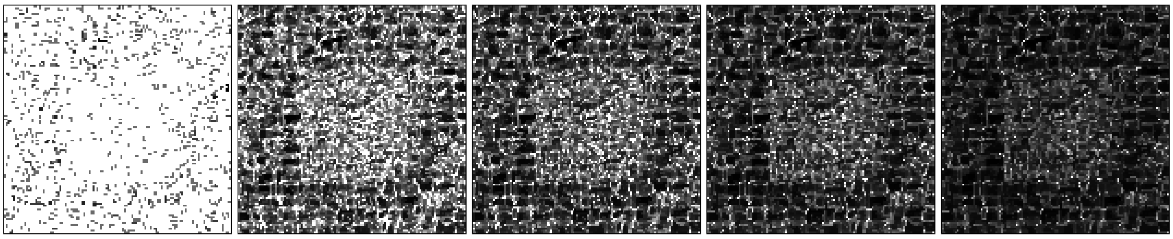


Fig. 4.6: Average inverse scale obtained with the region-merging algorithm.
FROM LEFT TO RIGHT:
(a) $\alpha_{Stop} = 1$. (b) $\alpha_{Stop} = 100$. (c) $\alpha_{Stop} = 250$. (d) $\alpha_{Stop} = 500$. (e) $\alpha_{Stop} = 1000$.

scales. This is favorable, since differences between small scales are more important for texture discrimination.

Fig. 4.6 and Fig. 4.7 show the average inverse scale for two test images. Bright regions thereby correspond to small scales, whereas dark regions indicate large scales. For better visibility the results have been normalized to a range between 0 and 255.

Fig. 4.6 displays the average inverse scale for the input image depicted in Fig. 4.5. In this image, a texture has been downsampled by factor 2 and copied back into the original texture. So the scale of the outer region is exactly twice the scale of the inner one. The result in Fig. 4.6a clearly shows that it is *not* sufficient to measure the scale directly in the original image without any simplification. For very small α almost all pixels are just considered to be single regions and so their scale is measured to be 1. The best results with regard to texture discrimination can be obtained with a value of α_{Stop} between 250 and 500. For such α most pixels have merged to small scale regions. Larger α obviously do not yield results that are useful for texture discrimination anymore.

Fig. 4.7 shows the inverse scale and the *average* inverse scale for the zebra test image in Fig. 4.2. One can see that although the very small scale texture can be distinguished from the rest of the image, the results are quite noisy. This is mainly due to the tendency of the region merging algorithm to keep single pixels even for quite large α if their value is considerably different from its neighbors. It can further be observed that the averaging has a small positive effect in this respect, because larger α can be considered without completely loosing the small scale history of pixels merged to larger regions.

Although the scale measure based on region merging gives some indication on the local scale and is quite easy to compute, it does not yield the quality one might have expected. The regions in the simplified images created by region merging are apparently not optimal for this kind of application. It is therefore reasonable to consider other simplification methods.

Scale measure based on TV flow. The second technique for measuring the local scale is based on the scale space created by TV flow. It has been shown in Chapter 2 that TV flow yields piecewise constant segmentation-like results.

However, while the region merging algorithm naturally determines the size of regions, TV flow is a diffusion technique and so the regions as well as their size are not explicit.
In order to measure the size of the emerging regions efficiently, the special properties of TV flow are exploited. As stated in Section 2.1.1, it can be proven in 1-D that pixels change their value inversely proportional to the size of the region they belong to. This allows to measure the size of the regions created by TV flow from the evolution speed of the pixels.

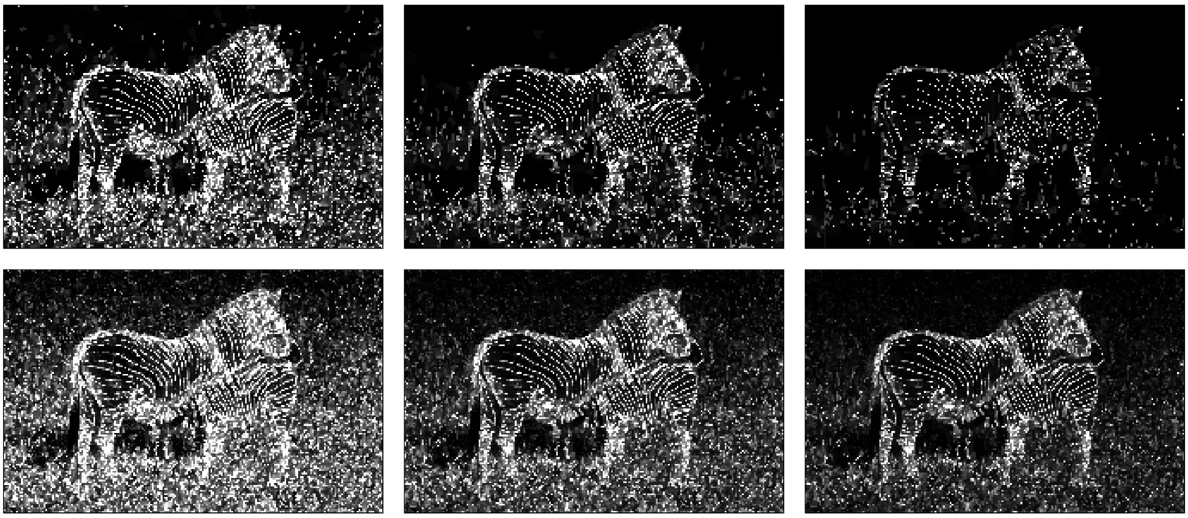


Fig. 4.7: Scale measure obtained with the region-merging algorithm for the zebra image in Fig. 4.2.
FROM LEFT TO RIGHT:
TOP ROW: Inverse scale for $\alpha = 100$, $\alpha = 250$, and $\alpha = 500$.
BOTTOM ROW: Average inverse scale for $\alpha_{Stop} = 100$, $\alpha_{Stop} = 250$, and $\alpha_{Stop} = 500$.

As a reminder, the rules of one-dimensional, space-discrete TV flow are:

(i) A region of m neighboring pixels with the same value can be considered as one superpixel with mass m.

(ii) The evolution splits into merging events where pixels melt together to larger pixels.

(iii) Extremum pixels adapt their value to that of their neighbors with speed $\frac{2}{m}$.

(iv) The two boundary pixels adapt their value with half that speed.

(v) All other pixels do not change their value.

These rules lead to the very useful consequence that by simply sitting upon a pixel and measuring the speed with which it changes its value, it is possible to determine its local scale. As pixels belonging to small regions move faster than pixels belonging to large regions (iii), the rate of change of a pixel determines the size of the region it currently belongs to. Integrating this rate of change over the evolution time T and normalizing it with this time yields the average speed of the pixel, i.e., its average inverse scale in scale space:

$$\frac{1}{\bar{m}} = \frac{1}{2} \frac{\int_0^T |\partial_t u| \, dt}{T} \tag{4.6}$$

The integration has to be stopped after some time T, otherwise the interesting scale information will be spoiled by scale estimates stemming from heavily oversimplified versions of the image. The parameter T has a similar meaning as the parameter α in the region merging technique.

Since only extremum regions change their value, periods of image evolution in which a pixel is not part of such an extremum region have to be taken into account. This can be done by reducing T for normalization by the time at which the pixel does not move (v), leading to the following formula:

$$\frac{1}{\bar{m}} = \frac{1}{2} \frac{\int_0^T |\partial_t u| \, dt}{T - \int_0^T \mathbf{1}_{\{\partial_t u = 0\}} \, dt} \tag{4.7}$$

where $\mathbf{1}_{\{a=b\}} = 1$ if $a = b$, and 0 otherwise. Besides the estimation error for the boundary regions, where the scale is overestimated by a factor 2 (iv), this formula yields exact estimates of the region sizes without any explicit representation of the regions in the 1-D case.

In 2-D the topology of regions can become more complicated. A region can be an extremum in one direction and a saddle point in another direction. There are no general analytic rules for the behavior of TV flow in 2-D. Therefore, it is no longer possible to obtain an *exact* estimate of the region sizes without an explicit representation of the regions. However, assuming that TV flow in 2-D still approximates the rules that hold in 1-D, one can get a scale measure by

$$\frac{1}{\bar{m}} = \frac{1}{4} \frac{\int_0^T |\partial_t u| \, dt}{T - \int_0^T \mathbf{1}_{\{\partial_t u = 0\}} \, dt} \tag{4.8}$$

where the pre-factor $\frac{1}{4}$ is due to the 4 neighbors of a pixel in 2-D. Fig. 4.8 reveals that this approximation still yields very good scale measures, in particular much better than the scale measure obtained with the region merging approach. The only problem is that, due to the behavior of TV flow in 2-D and numerical errors, $\partial_t u$ is very small at non-extrema, but hardly equal 0. One therefore obtains bad estimates for pixels that belong to non-extrema for a very long time. However, these areas are so small in practice that they do in general not frustrate the correct distinction of two textured regions.

Fig. 4.8 shows the local scale measure for the zebra image in Fig. 4.2 and different stopping times T. It illustrates the influence of this parameter. For instance, in Fig. 4.8c the pixels of the grass texture have been part of the large background region for such a long time, that their small scale history has hardly any influence anymore. Mainly the stripes of the zebras stand out from the background. Although this is very good for extracting the zebras in this image, such large stopping times are not so well suited for small scale texture discrimination, as Fig. 4.9 reveals. Also if T is chosen too small, the result is not optimal, as mainly the scale of noise is measured, see Fig. 4.8a. and Fig. 4.9a,b. However, for reasonable T, as depicted in Fig. 4.8b and Fig. 4.9c, the scale measure is of significantly higher quality than the scale measure obtained with the region merging algorithm. Obviously TV flow creates a scale space that is better suited for measuring local scale than the Mumford-Shah functional.
Note that the results show the diffusion speed of the pixels, i.e. the *inverse* scale, so dark regions correspond to large scales. Further on, the values have again been normalized to a

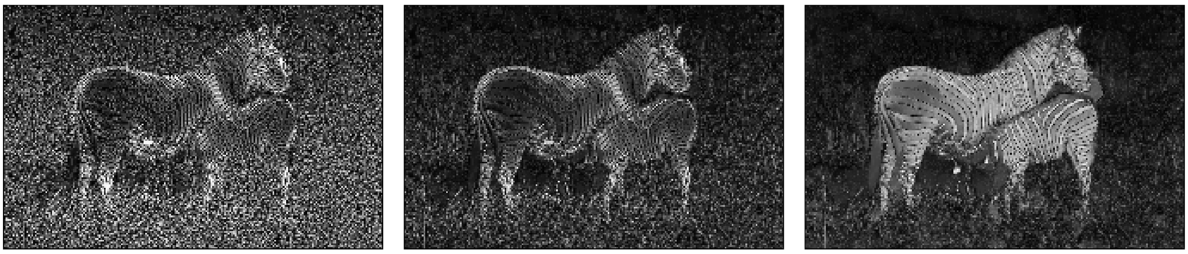


Fig. 4.8: Local scale measure based on TV flow for the zebra image in Fig. 4.2. The measure yields the inverse scale, so dark regions correspond to large scales, bright regions to fine scales. LEFT: (a) $T = 5$. CENTER:(b) $T = 20$. RIGHT: (c) $T = 100$.

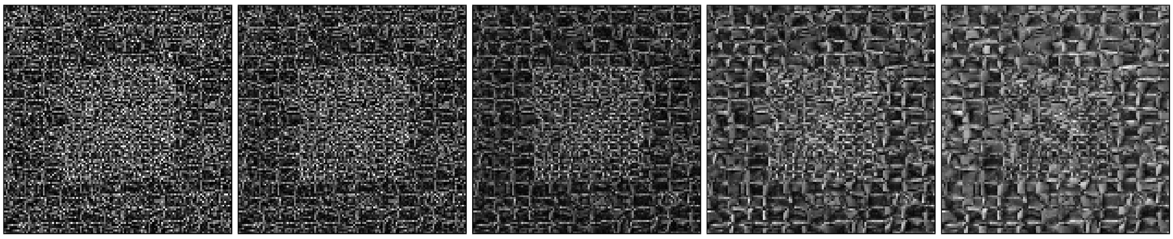


Fig. 4.9: Average inverse scale $\frac{1}{\bar{m}}$ obtained with TV flow for the image in Fig. 4.5. FROM LEFT TO RIGHT: (a) $T = 1$. (b) $T = 5$. (c) $T = 20$. (d) $T = 50$. (e) $T = 100$.

range between 0 and 255. Actually the results for higher T become significantly darker as the average scale increases.

4.1.3 A Sparse Set of Texture Features

Combining the structure tensor components (J_{11}, J_{22}, J_{12}) and the scale measure $\frac{1}{\bar{m}}$ from the previous two sections yields a texture feature vector

$$\mathbf{F} := \left(J_{11}, J_{22}, J_{12}, \frac{1}{\bar{m}} \right) \tag{4.9}$$

that basically contains the same information as the responses of a Gabor filter bank, though the feature space has been reduced to 4 dimensions.

Independent from the representation of the texture information, the quality of the texture features for the purpose of texture discrimination can be improved, if the feature space is smoothed by a coupled nonlinear diffusion process. This coupled smoothing deals with outliers in the data, propagates information into areas without filter responses, and synchronizes all channels, which eases further processing in a segmentation framework. Actually the components of the structure tensor have already undergone a nonlinear diffusion process, but since a coupling of all feature channels is desired and it is not reasonable to smooth the structure tensor components twice, they are replaced by the components of the *unsmoothed* structure tensor J_0.

The idea of smoothing a texture feature space by means of a discontinuity preserving technique has previously been proposed in [SSZ01], where Beltrami flow is used to smooth the responses of Gabor filters. In this thesis, we choose the nonlinear diffusion techniques provided in Section 2.1. Another option would be variational denoising as described in Section 2.3.

Using diffusion instead of variational denoising has two advantages. Firstly, edge enhancing flow (cf. Section 2.1.2) with $p > 1$ can be employed without loosing well-posedness in the discrete setting. The variational counterpart to edge enhancing flow, on the other hand, is based on a non-convex penalizer and therefore well-posedness questions are unresolved[1]. Edge enhancing flow is very well suited for smoothing texture features as it enhances edges and thus provides a kind of pre-segmentation that keeps the contrast much better than TV flow, cf. the examples in Section 2.1.2.

Secondly, a diffusion approach is well suited to apply certain automatic stopping criteria for the diffusion time t. This advantage will be exploited in Chapter 5.

[1] See Section 2.3.3

There is a further detail to be considered. The components of the structure tensor have a different dynamic range than the scale measure. Let Δ_I be the dynamic range of the input image I. Then if central differences are used for the gradient approximation, the components of the structure tensor have a dynamic range of $\frac{\Delta_I^2}{4}$. The values of the average inverse scale $\frac{1}{\tilde{m}}$, on the other hand, are between 0 and 1.

In order to ensure a fair coupling in the vector-valued diffusion scheme from (2.7), it is necessary that all feature channels have approximately the same dynamic range. Furthermore, the normalization procedure must not amplify the noise in case that one channel shows only a low contrast. Therefore, only normalization procedures that are independent from the contrast in the input data are applicable.

Anyway the squares in the structure tensor components are not optimal for discrimination purposes, since small gradients are almost neglected in comparison to large gradients. Therefore, and in order to ease the coupling with other features, the unsmoothed structure tensor J_0 is replaced by its square root. Given the eigenvalue decomposition $J_0 = U(\lambda_i)U^\top$ of the positive semidefinite and symmetric matrix, the square root can be computed by

$$\tilde{J}_0 := \sqrt{J_0} = U(\sqrt{\lambda_i})U^\top. \tag{4.10}$$

If J_0 is based on a scalar-valued input image I and not on a color image, the square root computation can be further simplified. In this case, J_0 has eigenvalues $|\nabla I|^2$ and 0 with corresponding eigenvectors $\frac{\nabla I}{|\nabla I|}$ and $\frac{\nabla I^\perp}{|\nabla I|}$, so $\tilde{J}_0$ comes down to

$$\tilde{J}_0 = \begin{pmatrix} \frac{I_x}{|\nabla I|} & -\frac{I_y}{|\nabla I|} \\ \frac{I_y}{|\nabla I|} & \frac{I_x}{|\nabla I|} \end{pmatrix} \begin{pmatrix} |\nabla I| & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{I_x}{|\nabla I|} & \frac{I_y}{|\nabla I|} \\ -\frac{I_y}{|\nabla I|} & \frac{I_x}{|\nabla I|} \end{pmatrix} = \begin{pmatrix} \frac{I_x^2}{|\nabla I|} & \frac{I_x I_y}{|\nabla I|} \\ \frac{I_x I_y}{|\nabla I|} & \frac{I_y^2}{|\nabla I|} \end{pmatrix} = \frac{J_0}{|\nabla I|}. \tag{4.11}$$

Further multiplication with 2 for compensating the central differences normalizes the structure tensor components to the same dynamic range as the input image. For the scale measure the normalization can be achieved by multiplying the values with Δ_I. Thus all channels have the same weight in a discontinuity preserving smoothing process or in a segmentation framework.

Using the initial condition $\mathbf{F}(0) = \left(2\tilde{J}_{0,11}, 2\tilde{J}_{0,22}, 4\tilde{J}_{0,12}, \frac{\Delta_I}{\tilde{m}}\right)$ and the diffusion process from (2.7), one obtains after diffusion time t the smoothed feature vector $\mathbf{F}(t)$. Note that the third structure tensor component has to be multiplied again by a factor 2 because it appears twice in the matrix.

The feature vector is depicted in Fig. 4.10 and Fig. 4.12 for two different input images. For comparison Fig. 4.11 and Fig. 4.13 show the smoothed Gabor feature space. Note that this time the feature space is depicted as it is, without normalization, in order to show that all channels have a very similar range of values.

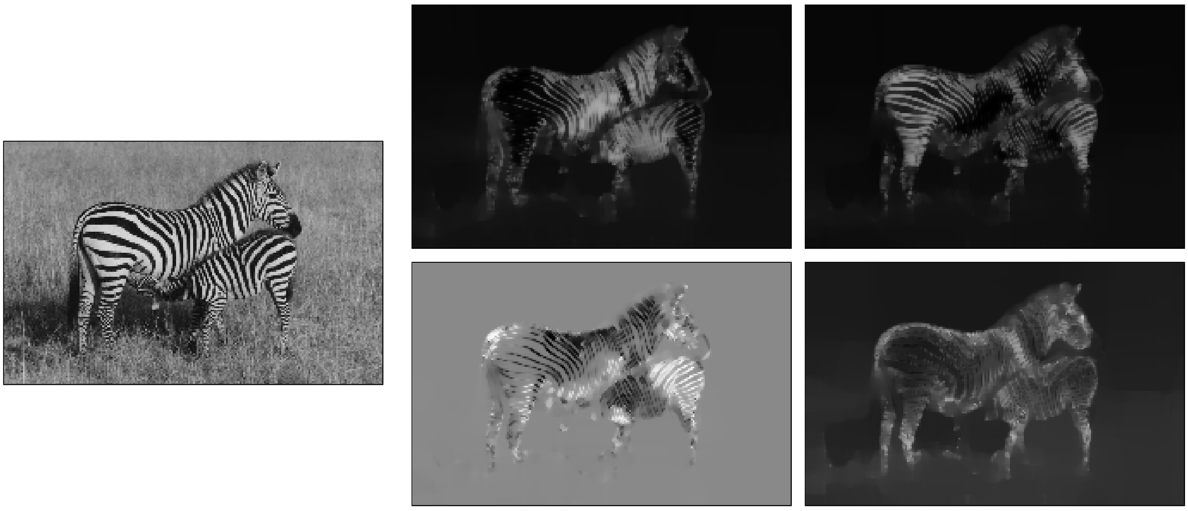


Fig. 4.10: LEFT: Input image (220×140). RIGHT: Sparse texture feature space $\mathbf{F}_t$ with $t = 500$. Components of $\mathbf{F}_t$ are shown from left to right, top to bottom, i.e., bottom right is the scale measure.

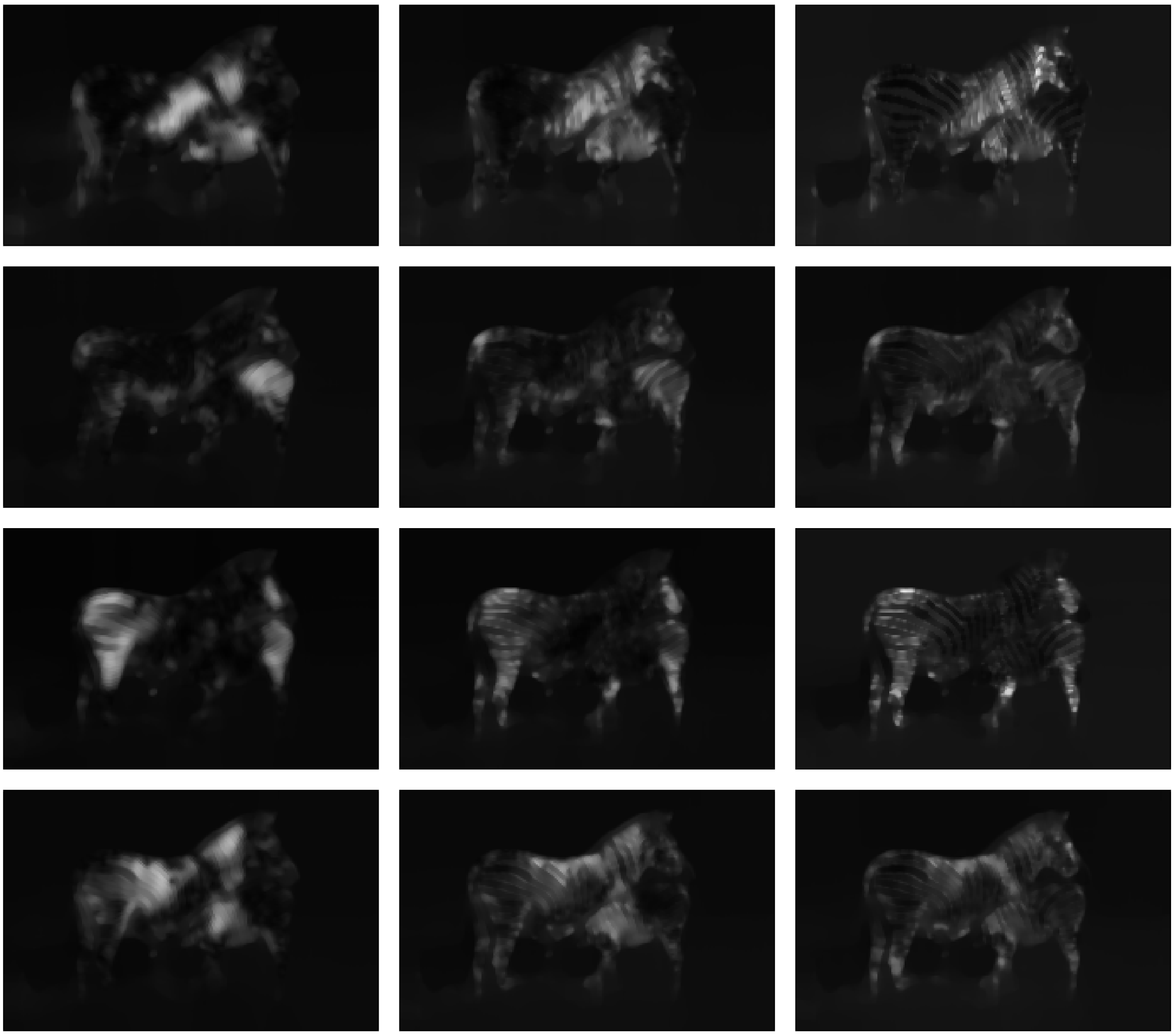


Fig. 4.11: Gabor feature space from Fig. 4.3 smoothed with $t = 500$. Same order as in Fig. 4.1 from left to right, top to bottom.

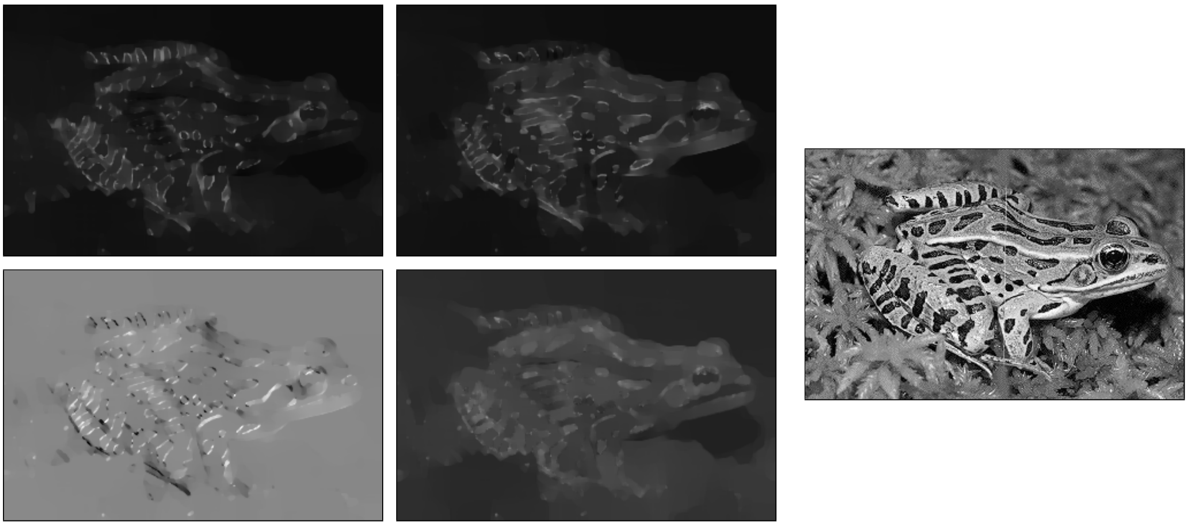


Fig. 4.12: LEFT: Sparse texture feature space $\mathbf{F}_t$ with $t = 1000$. RIGHT: Input image (329×220). Components of $\mathbf{F}_t$ are shown from left to right, top to bottom, i.e., bottom right is the scale measure.

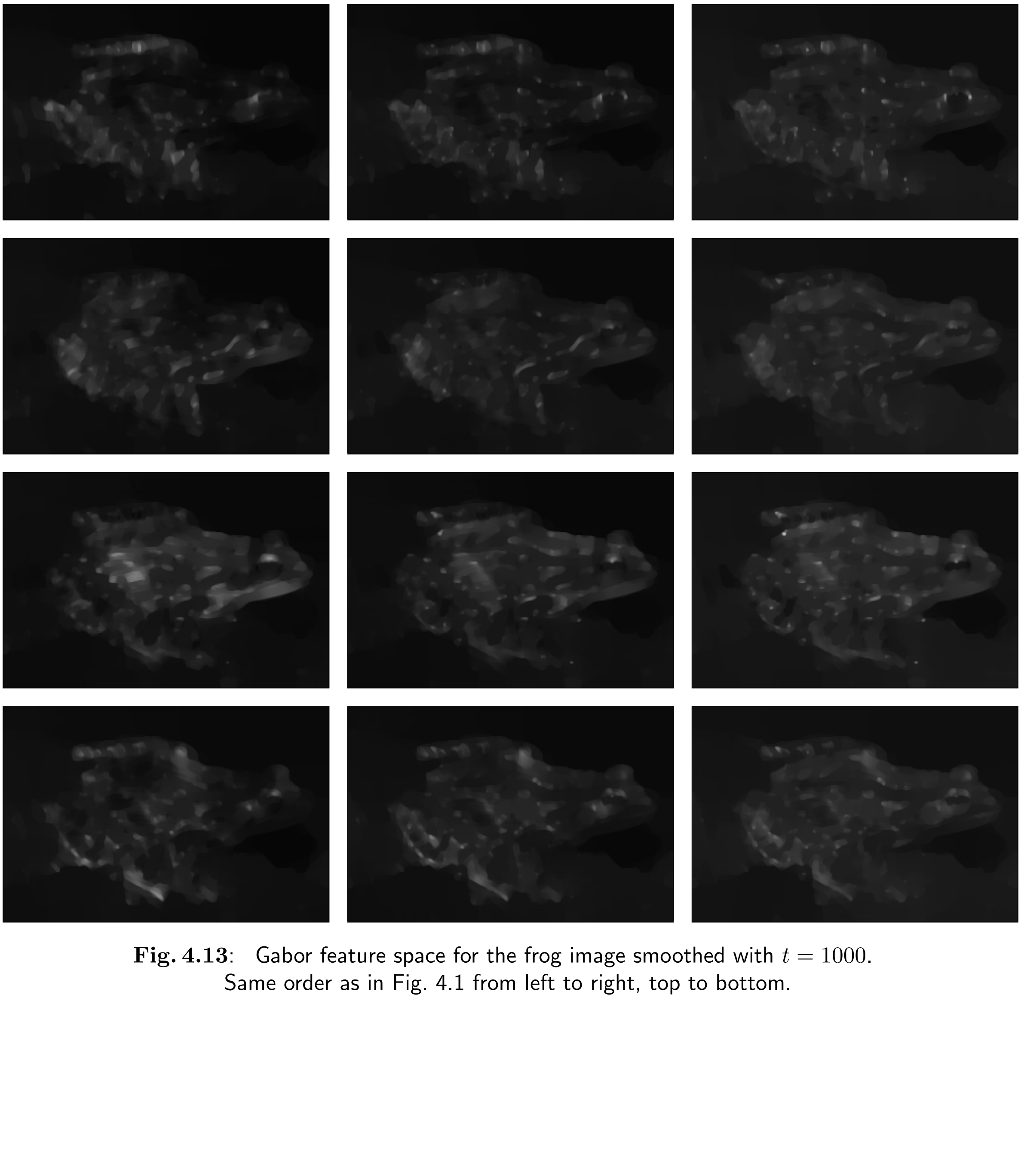


Fig. 4.13: Gabor feature space for the frog image smoothed with $t = 1000$. Same order as in Fig. 4.1 from left to right, top to bottom.

In the next chapter the feature space $\mathbf{F}(t)$ will be an important part of the segmentation framework described there. For segmentation both good discrimination capabilities and a good localization accuracy are important.

For the moment the texture features are evaluated in a much simpler experiment where only the discrimination capabilities play a role, as the textures are already separated from each other. Fig. 4.14 shows 8 different textures from the photographic album of Brodatz [Bro66] and the dissimilarities measured between these textures with the Gabor feature space as well as our sparse feature space.

As measure of dissimilarity for each feature channel a simple distance measure taking into account the means $\mu_k(T)$ and the standard deviations $\sigma_k(T)$ of each feature channel k of two textures T_1 and T_2 has been chosen:

$$\Delta_k = \left(\frac{\mu_k(T_1) - \mu_k(T_2)}{\sigma_k(T_1) + \sigma_k(T_2)} \right)^2 . \tag{4.12}$$

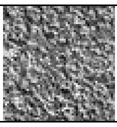
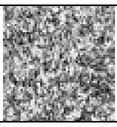
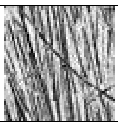
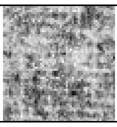
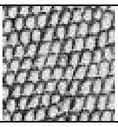
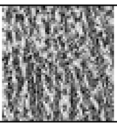
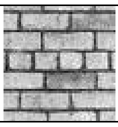
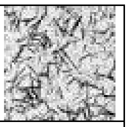
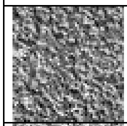
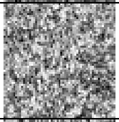
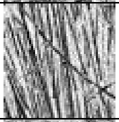
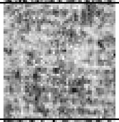
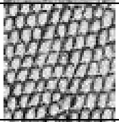
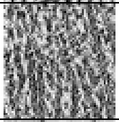
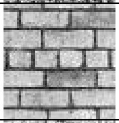
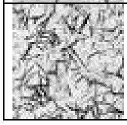
								
	0 0	285 88	1067 339	915 387	834 123	466 64	1496 629	391 416
	285 88	0 0	1106 180	971 270	367 104	205 43	1734 576	73 142
	1067 339	1106 180	0 0	1354 395	1450 275	620 130	1924 803	1005 268
	915 387	971 270	1354 395	0 0	1455 287	1363 287	305 159	672 233
	834 123	367 104	1450 275	1455 287	0 0	403 100	2013 469	355 183
	466 64	205 43	620 130	1363 287	403 100	0 0	2248 720	335 275
	1496 629	1734 576	1924 803	305 159	2013 469	2248 720	0 0	1222 426
	391 416	73 142	1005 268	672 233	355 183	335 275	1222 426	0 0

Fig. 4.14: Dissimilarities measured between some Brodatz textures.
FIRST VALUE: Dissimilarity in the Gabor feature space.
SECOND VALUE: Dissimilarity in the sparse feature space.

For the total dissimilarity the average of all M texture channels is computed:

$$\Delta = \frac{1}{M} \sum_{k=1}^{M} \Delta_k. \tag{4.13}$$

Additionally to the 12, respectively 4, texture channels, also the gray value is added as supplementary feature. Otherwise the results would not be in accordance with the visual impression when two textures have different average gray values. Smoothing would only reduce the standard deviation of the dissimilarity measure in this experiment. It has no influence on the mean values. Thus both feature spaces have not been smoothed here by nonlinear diffusion, i.e. $t = 0$.

The first value in Fig. 4.14 indicates the dissimilarity computed in the Gabor feature space, while the second value shows the distance in the sparse feature space. With both feature spaces, the computed values are in accordance with what one would expect from a measure of texture dissimilarity. Mostly the value of the Gabor feature space is around three or four times larger than the value of the sparse feature space, so up to a constant factor both feature spaces yield approximately the same values indicating that indeed they discriminate textures by means of the same properties. Only in a few cases there is a significant discrepancy. In such cases it is difficult to decide which feature space is more in accordance with the expectation. This gives much evidence that, instead of the 12-dimensional feature space also the sparse, and more efficient 4-dimensional feature space can be used for texture segmentation without compromising the quality of the segmentation results. This is further supported by the segmentation results shown in Chapter 5.

4.2 Optic Flow

We now consider optic flow estimation as one of the primary research topics in computer vision. It is seeking the displacement field that describes the pixel shift between two images. This correspondence problem appears quite frequently in computer vision, once in motion estimation and once in stereo vision.

Optic flow can be a useful cue for image segmentation, as it allows the separation of a moving object from a static or differently moving background. Speaking from one's own experience, it can be very difficult to detect a camouflage animal when it sits still, yet all the camouflage is gone as soon as the animal moves.

Goal is the computation of the optic flow vector $\mathbf{w}(\mathbf{x}) := (u(\mathbf{x}), v(\mathbf{x}), h_t)$ between two successive images of an image sequence $I : (\Omega \subset \mathbb{R}^3) \to \mathbb{R}$. where $\mathbf{x} = (x, y, z)$ is now a point in the spatio-temporal domain with z denoting the time axis. The grid size in temporal direction is set to be $h_t = 1$. In order to estimate this flow vector, it is necessary to define a constancy assumption, i.e., a property of the pixels has to be assumed not to change from one image to the next. A natural choice is the image gray value. Assuming that it stays constant between the pixel (x, y) in the image at time z and the shifted pixel $(x+u, y+v)$ in a successive image at time $z + 1$ yields the constraint

$$I(x, y, z) = I(x + u, y + v, z + 1). \tag{4.14}$$

This is a nonlinear equation in u and v. In order to resolve for u and v the equation is mostly linearized by a first order Taylor expansion. This leads to the linearized grey value constancy assumption, the so-called *optic flow constraint (OFC)* [HS81]:

$$I_x u + I_y v + I_z = 0 \tag{4.15}$$

where subscripts denote partial derivatives. It must be noted that this linearization is only valid under the assumption that the image changes linearly along the displacement, which is in general not the case, especially for large displacements. In Section 4.2.3 it is therefore proposed to compute the optic flow from the original nonlinear equation (4.14). For the moment, however, we restrict ourselves to the much simpler OFC in (4.15).

Having instead of a sequence of gray value images $I(x, y, z)$ a sequence of multichannel images, e.g. color images with the color channels R, G, and B, it is reasonable to assume that the values in all channels stay constant. So instead of one equation, a system of equations is obtained

$$\begin{aligned} R(x, y, z) &= R(x + u, y + v, z + 1) \\ G(x, y, z) &= G(x + u, y + v, z + 1) \\ B(x, y, z) &= B(x + u, y + v, z + 1) \end{aligned} \tag{4.16}$$

which can as well be linearized to become a linear system of equations:

$$\begin{aligned} R_x u + R_y v + R_z &= 0 \\ G_x u + G_y v + R_z &= 0 \\ B_x u + B_y v + R_z &= 0. \end{aligned} \tag{4.17}$$

From (4.15) it is easy to see that one assumption is not sufficient to determine a unique solution for the optic flow, as there is only one equation for two unknowns. With only the gray value constancy assumption, one can merely determine the so-called *normal flow* in direction of the gradient. In the case of a color sequence it looks on the first glance like if there is enough information to compute a unique solution. However, since the gradient directions of the different color channels are in many cases quite similar, the smaller eigenvalue of the system matrix can still be very small, thus the information is often not sufficient to compute a reliable estimate of u and v.

Consequently, a second assumption is needed that is capable to provide a unique solution of the flow vector. There are two popular possibilities. The first goes back to the early work of Lucas and Kanade [LK81] and assumes that the optic flow in a local neighborhood can be described by a parametric model, which is in the simplest case the model of constant flow. This allows to *locally* compute the optic flow for each pixel ignoring the situation outside the local neighborhood. Hence, these methods are also called *local methods*.
The other class of techniques is based on the work of Horn and Schunck [HS81] and assumes the optic flow field to be smooth. Since this induces a dependency of the flow vector at a pixel on the flow at all other pixels, these methods are often called *global methods*.

4.2.1 Local Methods

In the original Lucas-Kanade method, the optic flow vector is simply assumed to be constant. For the neighborhood one often uses a Gaussian window K_ρ with standard deviation ρ. This allows to estimate the optic flow in each point as the minimizer of the energy function

$$E(u,v) = \frac{1}{2} K_\rho * \left((I_x u + I_y v + I_z)^2 \right). \tag{4.18}$$

The convolution with the Gaussian can thereby be performed in the spatial as well as in the spatio-temporal domain, as will be discussed later on. A minimum (u,v) of E satisfies $\partial_u E = 0$ and $\partial_v E = 0$, leading to the linear system

$$\begin{pmatrix} K_\rho * I_x^2 & K_\rho * I_x I_y \\ K_\rho * I_x I_y & K_\rho * I_y^2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -K_\rho * I_x I_z \\ -K_\rho * I_y I_z \end{pmatrix} \tag{4.19}$$

Provided the neighborhood defined by K_ρ is large enough to capture gradients of significantly different direction, both eigenvalues of the system matrix are considerably larger than 0 and the linear system can easily be resolved for u and v.

Local methods involve the structure tensor. Obviously, the entries of this linear system are five of the six different components of the spatio-temporal linear structure tensor[2]

$$J_\rho = K_\rho * \left(\nabla I \nabla I^\top \right) = K_\rho * \begin{pmatrix} I_x^2 & I_x I_y & I_x I_z \\ I_x I_y & I_y^2 & I_y I_z \\ I_x I_z & I_y I_z & I_z^2 \end{pmatrix}. \tag{4.20}$$

Indeed, local optic flow estimation can also be considered as finding the spatio-temporal orientation with least change in the image sequence [BGW91]. This means minimizing

$$E(\mathbf{v}) = \mathbf{v}^\top J_\rho \mathbf{v} \tag{4.21}$$

under the constraint $|\mathbf{v}| = 1$. So the minimum $\mathbf{v}$ is the eigenvector corresponding to the smallest eigenvalue of the structure tensor J_ρ. The optic flow vector $\mathbf{w}$ can then be obtained by normalizing $\mathbf{v}$ such that the third component is 1.

[2] See also Section 2.2.

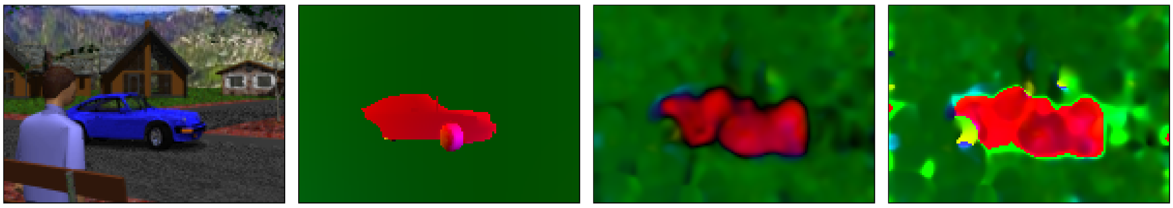


Fig. 4.16: STREET sequence ($140 \times 100 \times 20$).
FROM LEFT TO RIGHT: (a) Frame 10. (b) Correct flow field between frame 10 and 11. (c) Result with ordinary least squares. (d) Result with total least squares.

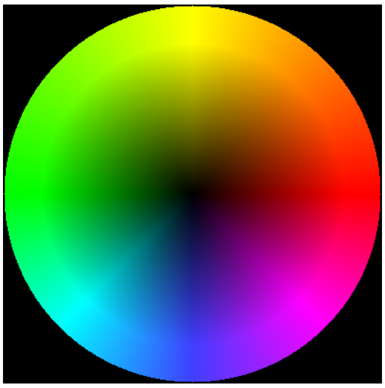


Fig. 4.15: Color code.

Although both ways to compute the optic flow vector from a local neighborhood are very similar and are both connected with the structure tensor, there is a difference. That is (4.19) computes an ordinary least squares estimate, whereas the minimization of (4.21) comes down to total least squares [vHV91]. While ordinary least squares tend to underestimate the length of the flow vector in situations with high ambiguities (e.g. at motion discontinuities), total least squares do not have this bias, yet the optic flow estimates are often much too large, in particular when the estimated orientation is wrong. This can be seen in Fig. 4.16 that depicts one frame of the STREET sequence[3], the correct flow field, and the flow fields computed with ordinary least squares and total least squares. Like for the visualization of the structure tensor, a color plot is used to show both orientation and magnitude of the optic flow vectors. The vector's orientation is indicated by the color, whereas the vector's magnitude is expressed by the intensity of the image. Fig. 4.15 shows which color corresponds to which orientation. Since the ordinary least squares approach supplies more reliable results, it will be used for the remainder of this section.

Extensions. Local methods have been extended in several ways. One improvement replaces the quadratic error measure by robust statistics [BA91, BHS98, YD99], another is the usage of parametric models of higher order, e.g. affine models [BA96, Far00]. These extensions are not discussed here. Instead, the integration of color information, the extension of the parametric assumption to the spatio-temporal domain [BGW91, BA91, Far00], as well as adaptive local neighborhoods [NG98, BW02, Bro02, LCR03, LKL$^+$04] are investigated.

Color can be included in optic flow computation by simply considering all channels of a color sequence $\mathbf{I} = (I_1, I_2, I_3) = (R, G, B)$ in the energy

$$E(u,v) = \frac{1}{2} K_\rho * \sum_{k=1}^{M} \Big((I_k)_x u + (I_k)_y v + (I_k)_z \Big)^2 . \tag{4.22}$$

This comes down to replacing the system entries in (4.19) by the components of the structure tensor for vector-valued images in (2.29). In general, the optic flow estimates become better when including color, because additional information can be exploited.

[3] This is a cropped version of a sequence that has been introduced in [GMN$^+$98] and is available at `www.cs.otago.ac.nz/research/vision`.

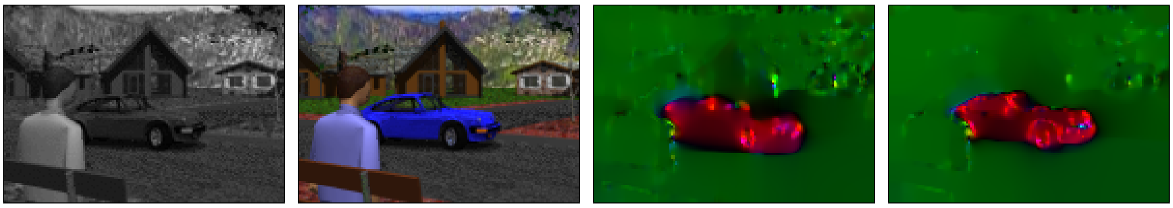


Fig. 4.17: Influence of color information.
FROM LEFT TO RIGHT:
(a) Frame 10 of the gray-valued STREET sequence. (b) Frame 10 of the colored sequence.
(c) Flow estimated in the gray value sequence. (d) Flow estimated in the color sequence.

The strategy to use additional information is also applied when the constancy assumption is extended to the spatio-temporal domain. This only needs the convolution with a spatio-temporal Gaussian instead of a spatial one, and in general yields improved results due to the information exchange along the temporal axis. However, it should be noted that the exploitation of this information leads also to additional violations of the constancy assumption. Thus for the sake of a higher robustness under noise and other artifacts, one introduces an artificial motion blur in the resulting flow field. This can become problematic for motion segmentation, in particular when large displacements are involved.

Violations of the flow constancy assumption are a severe problem of the Lucas-Kanade method. Especially near motion discontinuities, they lead to bad optic flow estimates. A possibility to avoid the violation of this assumption is the replacement of the simple non-adaptive Gaussian neighborhood by a data-adaptive neighborhood. This has been proposed in [NG98], where an adaptive anisotropic Gaussian is used as local neighborhood.
A better adaptivity can, however, be obtained by means of nonlinear diffusion [BW02, Bro02] and the nonlinear structure tensor described in Section 2.2.2[4]. If the entries in the linear system in (4.19) are replaced by the corresponding components of the nonlinear structure tensor, one obtains a flow estimation method for which the constant flow assumption should hold in the neighborhood determined by the discontinuities in the structure tensor. As this is much more often true than the constant flow assumption in a fixed Gaussian neighborhood, one can expect more accurate estimation results especially near motion discontinuities.

Experiments. The differences between the diverse Lucas-Kanade variants are uncovered in several experiments, the results of which are depicted in Fig. 4.17 - Fig. 4.21. Whereas these figures give a good visual impression of the results, the fact that the correct flow field is available for all test sequences offers also the possibility to have a quantitative measure for the quality of the estimated optic flow.

Such a quantitative quality measure has been introduced with the so-called *average angular error (AAE)* in [BFB94]. Assumed the correct flow field $\mathbf{w_c} = (u_c, v_c, 1)$ is known, which is the case for several synthetic image sequences, the AAE of the estimated flow field $\mathbf{w}$ can be computed by:

$$\text{AAE} := \frac{1}{n} \sum_{i=1}^{n} \arccos \left(\frac{(u_c)_i u_i + (v_c)_i v_i + 1}{\sqrt{((u_c)_i^2 + (v_c)_i^2 + 1)(u_i^2 + v_i^2 + 1)}} \right). \tag{4.23}$$

where n denotes the total number of pixels. Against its indication, this expression not only measures the spatial angular error between the estimated flow vector and the correct vector,

[4] See also the relations between adaptive structure tensors in Section 2.2.2.

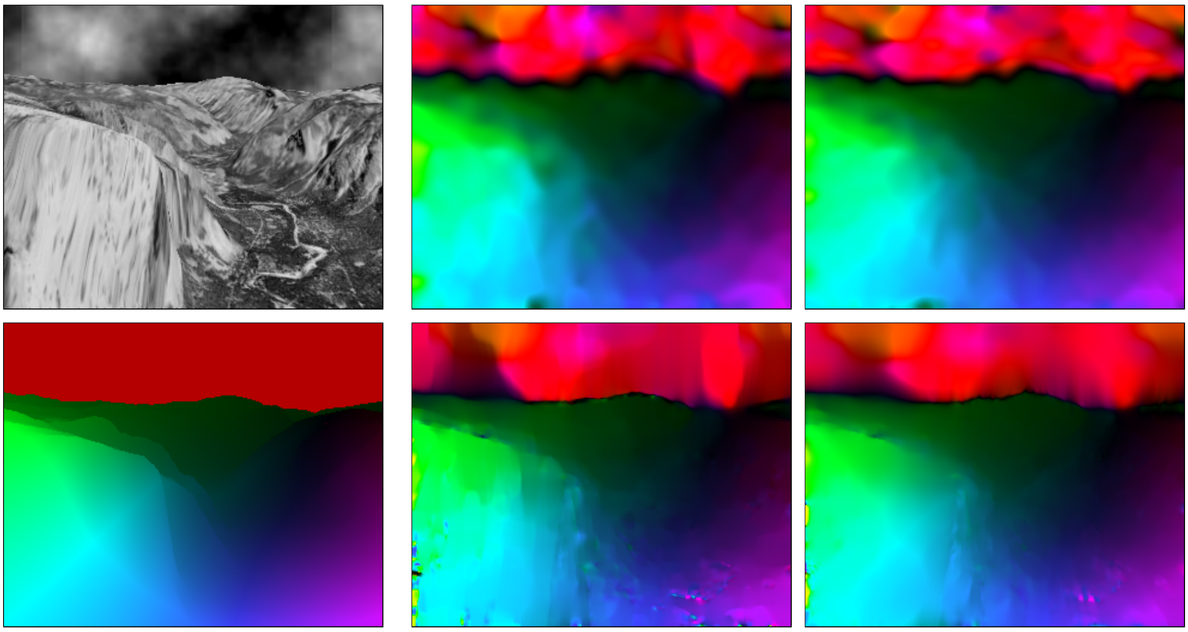


Fig. 4.18: YOSEMITE sequence with cloudy sky ($316 \times 252 \times 15$)
TOP LEFT: Frame 8 of the sequence. BOTTOM LEFT: Correct flow field between frame 8 and 9.
TOP MIDDLE: Linear Lucas-Kanade. TOP RIGHT: Linear Lucas-Kanade 3-D.
BOTTOM MIDDLE: Nonlinear Lucas-Kanade. BOTTOM RIGHT: Nonlinear Lucas-Kanade 3-D.

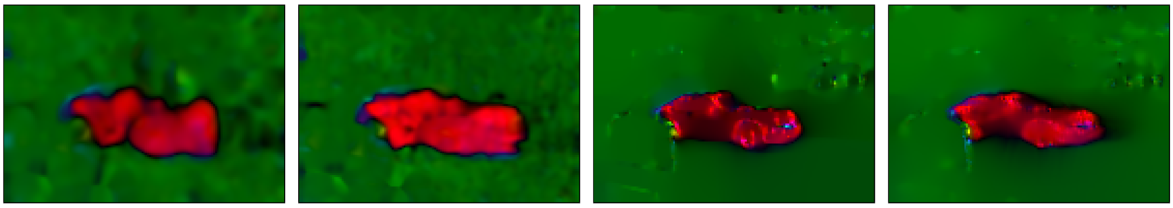


Fig. 4.19: STREET sequence ($140 \times 100 \times 20$).
FROM LEFT TO RIGHT: (a) Linear Lucas-Kanade. (b) Linear Lucas-Kanade 3-D.
(c) Nonlinear Lucas-Kanade. (d) Nonlinear Lucas-Kanade 3-D.

but also differences in the magnitude of both vectors, since it evaluates the angular error of the *spatio-temporal* vector $(u, v, 1)$.

In a first experiment, the influence of color information has been tested. For this purpose the methods using the nonlinear structure tensor have been compared, since the method with an adaptive neighborhood is supposed to benefit more from the additional information than the method with the fixed neighborhood. Indeed one can see a clear difference between the two results depicted in Fig. 4.17. This is confirmed also by the AAE indicated in Tab. 4.1. Besides the AAE, this table also shows the optimized parameters and the standard deviation of the angular error.

Fig. 4.18 - Fig. 4.20 reveal the effect of the spatio-temporal constancy assumption as well as the adaptive neighborhood due to the nonlinear structure tensor[5]. It can be seen that

[5] The YOSEMITE sequence has been created by Lynn Quam and is available at `ftp://ftp.csd.uwo.ca/pub/vision`. The MARBLE sequence has been created by Otte and Nagel and can be obtained from `http://i21www.ira.uka.de/image_sequences`. Note that the black areas in the correct flow field are not considered for the computation of the average angular error, since there is no ground truth available.

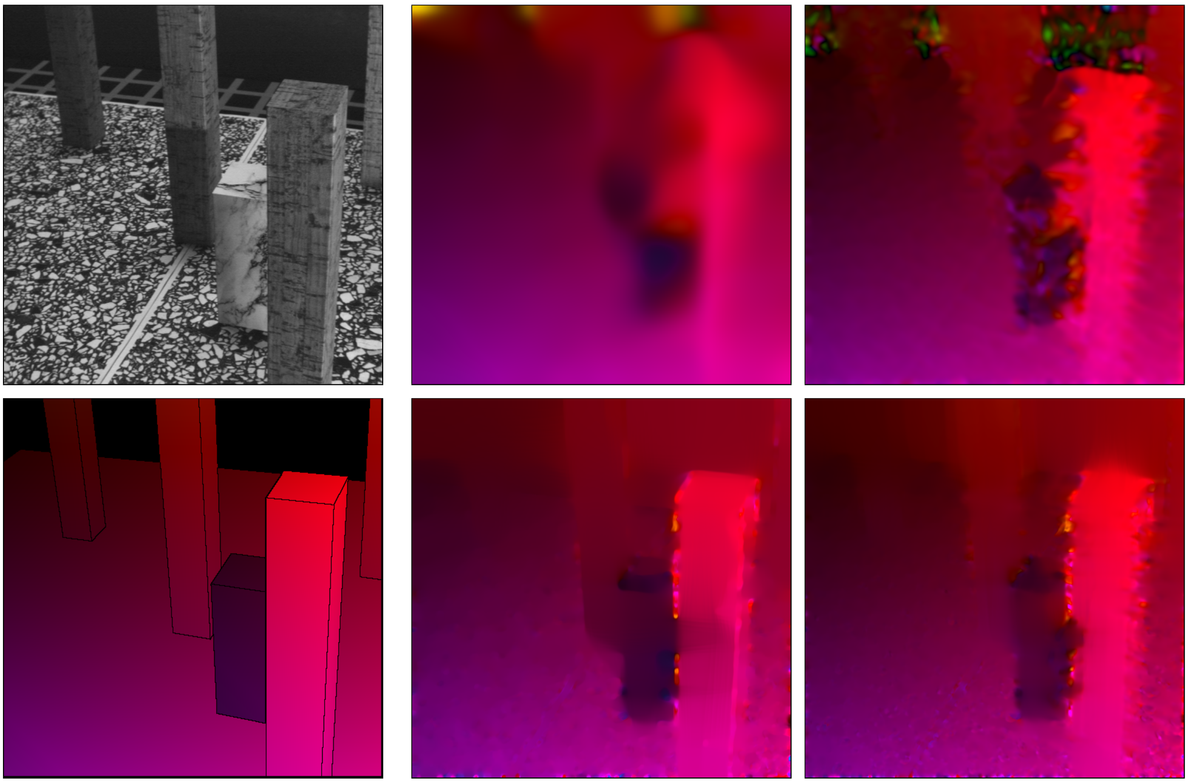


Fig. 4.20: MARBLE sequence ($512 \times 512 \times 31$).
TOP LEFT: Frame 16 of the sequence. BOTTOM LEFT: Correct flow field between frame 16 and 17.
TOP MIDDLE: Linear Lucas-Kanade. TOP RIGHT: Linear Lucas-Kanade 3-D.
BOTTOM MIDDLE: Nonlinear Lucas-Kanade. BOTTOM RIGHT: Nonlinear Lucas-Kanade 3-D.

the latter always leads to much more accurate results especially near motion discontinuities. Also the quantitative measure confirms the general positive effect of the nonlinear structure tensor. The AAE of the adaptive method is always significantly smaller than the AAE of the classic method. For the YOSEMITE sequence, Tab. 4.1 also contains the value of the nonlinear structure tensor proposed in [BW02, Bro02]. With the nonlinear structure tensor based on TV flow, again a significant improvement can be effectuated.

Also the spatio-temporal neighborhood has mostly positive effects on the result. This is in particular true, if the motion is approximately the same in each frame of the sequence, as this is the case for the YOSEMITE sequence and the MARBLE sequence. If, on the other hand, the motion constancy assumption over time is not true, as this is the case for the STREET sequence, the purely spatial neighborhood can yield favorable results. Here the artificial motion blur induced by the spatio-temporal neighborhood compromises the accuracy at motion discontinuities.

Fig. 4.21 finally shows the influence of the edge enhancement parameter p in the nonlinear structure tensor. While $p = 0.8$ leads to results that are closer to the classic Lucas-Kanade method, $p = 1.2$ yields very sharp edges and a segmentation-like flow field. However, one can see in Tab. 4.1 that such sharp edges do not necessarily lead to the most accurate optic flow results. The experiment indicates that for optic flow estimation TV flow seems to be the best choice to smooth the structure tensor.

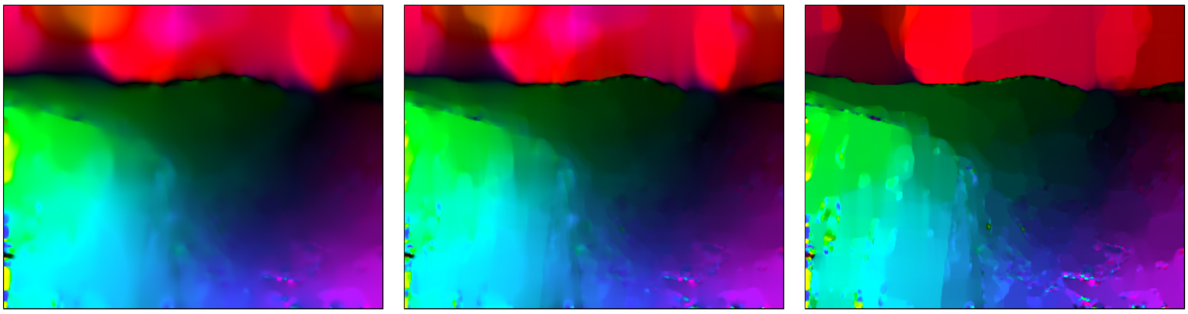


Fig. 4.21: Comparison of results obtained with the nonlinear structure tensor and different amounts of edge preservation/enhancement.
Left: $p = 0.8$. Center: $p = 1$ (TV flow). Right: $p = 1.2$.

Yosemite sequence.

Technique	p	σ	σ_z	t	AAE	Std. dev.
Linear structure tensor	0	1.4	-	21	8.78°	±12.76°
Nonlinear structure tensor from [Bro02]	-	1.2	-	135	8.22°	±12.59°
Nonlinear structure tensor	0.8	1.3	-	240	8.02°	±12.63°
Nonlinear structure tensor	1.2	1.0	-	1200	7.90°	±10.60°
Nonlinear structure tensor	1	1.3	-	400	7.65°	±11.05°
Linear structure tensor 3-D	0	1.3	0.6	12	7.60°	±12.11°
Nonlinear structure tensor from [Bro02] 3-D	-	1.2	0.7	18	7.49°	±12.53°
Nonlinear structure tensor 3-D	1	1.3	0.6	140	6.82°	±10.62°

Street sequence.

Technique	p	σ	σ_z	t	AAE	Std. dev.
Linear structure tensor, total least squares	0	0.9	-	3	14.94°	±26.41°
Linear structure tensor	0	0.7	-	3.25	10.54°	±19.48°
Linear structure tensor 3-D	0	0.9	0.3	1.25	10.46°	±22.10°
Nonlinear structure tensor, gray value	1	0.6	-	275	9.81°	±15.52°
Nonlinear structure tensor 3-D	1	0.4	0.1	1900	8.76°	±13.49°
Nonlinear structure tensor	1	0.4	-	3000	7.75°	±12.50°

Marble sequence.

Technique	p	σ	σ_z	t	AAE	Std. dev.
Linear structure tensor	0	2.5	-	220	5.87°	±4.45°
Nonlinear structure tensor	1	2.5	-	250	5.26°	±3.06°
Linear structure tensor 3-D	0	2.5	0.3	10	3.71°	±5.63°
Nonlinear structure tensor 3-D	1	2.5	0.3	100	3.24°	±4.22°

Tab. 4.1: Quantitative evaluation of the methods. AAE = average angular error. The parameters σ and σ_z denote the standard deviation of the Gaussian kernel used for pre-smoothing the image sequence in spatial and temporal direction, respectively.

4.2.2 Global Methods

In contrast to local optic flow estimation methods, global methods are based on a global smoothness assumption for the estimated flow field. Such an assumption is beneficial, as the true optic flow often does not fit to a certain parametric model. Although a global smoothness assumption seems to yield a much harder problem to solve, the possibility to formulate it as a variational approach makes this problem very well tractable.

An early global approach to optic flow estimation has been proposed by Horn and Schunck [HS81]. Using the optic flow constraint as data fidelity term and the Tikhonov regularizer as smoothness term, it is described by the energy functional:

$$E(u,v) = \int_\Omega \left(\underbrace{(I_x u + I_y v + I_z)^2}_{\text{Data}} + \alpha \underbrace{(|\nabla u|^2 + |\nabla v|^2)}_{\text{Smoothness}} \right) \mathbf{dx}. \tag{4.24}$$

Comparing this energy functional with (2.40) for variational denoising in Section 2.3, one can see that it has the same structure. Only the data term differs, as the result should not be close to the input image, but should fulfill the optic flow constraint. The smoothness term reflects the search for a *vector*-valued minimizer. An optimum must satisfy the following two Euler-Lagrange equations:

$$\begin{array}{rcl} (I_x u + I_y v + I_z) I_x - \alpha \Delta u &=& 0 \\ (I_x u + I_y v + I_z) I_y - \alpha \Delta v &=& 0 \end{array} \tag{4.25}$$

with reflecting boundary conditions $\partial_n u = 0$ and $\partial_n v = 0$ on $\partial\Omega$. Like in Section 2.3.1, the discretization of these Euler-Lagrange equations leads to a sparse linear system of equations which can be solved by means of an iterative solver like Gauss-Seidel or SOR [You71]. Details on the numerics are given later in this section.

As local methods, also the global approach can easily be extended to vector-valued image sequences $\mathbf{I} = (I_1, ..., I_M)$ by considering all channels in the least squares setting

$$E(u,v) = \int_\Omega \left(\sum_{k=1}^{M} ((I_k)_x u + (I_k)_y v + (I_k)_z)^2 + \alpha(|\nabla u|^2 + |\nabla v|^2) \right) \mathbf{dx}. \tag{4.26}$$

The Euler-Lagrange equations then read:

$$\begin{array}{rcl} \sum_{k=1}^{M} \left((I_k)_x u + (I_k)_y v + (I_k)_z\right)(I_k)_x - \alpha \Delta u &=& 0 \\ \sum_{k=1}^{M} \left((I_k)_x u + (I_k)_y v + (I_k)_z\right)(I_k)_y - \alpha \Delta v &=& 0 \end{array} \tag{4.27}$$

Non-quadratic smoothness terms. It has already been mentioned in Section 2.3 that quadratic penalizers, such as the Tikhonov regularizer used in (4.24), give too much influence to outliers, since outliers are not accepted by the model. In order to respect outliers in the model assumptions, the quadratic penalizer is to be replaced by a robust function $\Psi(s^2)$, for instance $\Psi(s^2) = \sqrt{s^2}$, which yields the TV regularizer.

The usage of non-quadratic robust functions in the smoothness assumption has also been proposed for optic flow estimation [Coh93, Sch94b, BA96, MP96, Wei98b, ADK99, WS01a][6].

[6] Note that the approaches in [Coh93, BA96, ADK99] are not rotationally invariant, because they apply the robust function separately to ∇u and ∇v.

The energy functional with a robust function applied reads

$$E(u,v) = \int_{\Omega} \left((I_x u + I_y v + I_z)^2 + \alpha \, \Psi(|\nabla u|^2 + |\nabla v|^2) \right) \, \mathbf{dx} \tag{4.28}$$

and accepts outliers in the smoothness assumption, i.e. discontinuities in the optic flow field. Consequently, such a model contains the assumption of a piecewise smooth flow field.

Similar smoothness assumptions that also yield discontinuities in the estimated flow field have been proposed in [Nag83, NE86, Nag87, Sch91, Sny91, Sch93, AELS99]. The main difference of these methods to the functional in (4.28) is, however, the determination of flow discontinuities by means of discontinuities in the image sequence. As discontinuities in the image do not necessarily correspond to discontinuities in the flow field, these so-called *image-driven* methods can in general not achieve the quality of *flow-driven* methods. A nice overview on regularizers in optic flow estimation can be found in [WS01a].

A minimizer of (4.28) must satisfy the Euler-Lagrange equations

$$\begin{aligned} (I_x u + I_y v + I_z) I_x - \alpha \, \mathrm{div}\left(\Psi'(|\nabla u|^2 + |\nabla v|^2) \nabla u\right) &= 0 \\ (I_x u + I_y v + I_z) I_y - \alpha \, \mathrm{div}\left(\Psi'(|\nabla u|^2 + |\nabla v|^2) \nabla v\right) &= 0 \end{aligned} \tag{4.29}$$

where $\Psi'(s^2)$ denotes the derivative of Ψ with respect to its argument. After discretization, these equations yield a *nonlinear* system of equations, which can be solved by a fixed point iteration scheme, as described in Section 2.3.2.

At this point it should be mentioned that the energy functional in (4.28) has a unique minimizer, if the penalizer function $\Psi(s^2)$ is convex in s [WS01a], so one can be sure to find the global optimum by a suitable iterative scheme independent from the initialization. For non-convex penalizer functions, on the other hand, the energy functional may have multiple local minimizers, and the efficient search for the global optimum can only be performed by heuristic optimization methods, which cannot guarantee anymore to yield really the global optimum. Therefore, a regularized version of the TV penalizer $\Psi(|\nabla u|^2 + |\nabla v|^2) = \sqrt{|\nabla u|^2 + |\nabla v|^2 + \epsilon^2}$ with a small ϵ is a good choice for the penalizer function, since it offers rather sharp discontinuities and still ensures a unique solution.

Spatio-temporal smoothness assumption. It has been mentioned in Section 4.2.1 that the flow constancy assumption in local optic flow methods can be easily extended to the spatio-temporal domain. The same has also been proposed for the smoothness assumption in global methods [Nag90, EF98, WS01b]. This only necessitates the interpretation of the gradient operator in ∇u and ∇v as the spatio-temporal gradient operator $\nabla = (\partial x, \partial y, \partial z)^\top$. Like in local optic flow estimation methods, the extension to the spatio-temporal domain in general leads to better estimates due to the additional information exchange along the temporal axis.

Non-quadratic data terms. Outliers not only occur in the smoothness assumption but as well in the data assumption. It is easy to imagine situations where the optic flow constraint is not fulfilled, e.g. because of occlusion, brightness changes, or noise. Thus it makes sense to apply a robust function also to the data term. This idea is in line with robust statistics applied in local flow estimation methods, as proposed in [BA91, BHS98, YD99] and has been transferred to global optic flow methods, e.g. in [BA96, MP96, BWS05]. With a robust function applied to both the data term and the smoothness term, the energy functional reads:

$$E(u,v) = \int_{\Omega} \left(\Psi((I_x u + I_y v + I_z)^2) + \alpha \, \Psi(|\nabla u|^2 + |\nabla v|^2) \right) \, \mathbf{dx}. \tag{4.30}$$

In the corresponding Euler-Lagrange equations

$$\begin{aligned}\Psi'((I_x u + I_y v + I_z)^2)(I_x u + I_y v + I_z)I_x - \alpha \operatorname{div}(\Psi'(|\nabla u|^2 + |\nabla v|^2)\nabla u) &= 0 \\ \Psi'((I_x u + I_y v + I_z)^2)(I_x u + I_y v + I_z)I_y - \alpha \operatorname{div}(\Psi'(|\nabla u|^2 + |\nabla v|^2)\nabla v) &= 0\end{aligned} \tag{4.31}$$

the robust data term yields a further nonlinearity, which can be resolved the same way as the nonlinearity induced by the robust smoothness term. Moreover, if $\Psi(s^2)$ is convex in s, there exists a unique minimizer for the energy in (4.30) [HSSW02]. Note that it is not necessary to choose the same penalizer function for the data and the smoothness term. For simplicity reasons, however, both functions are restricted in this work to be equal.

Gradient constancy assumption. The basic assumption of most optic flow estimation methods, namely the assumption that the gray value of a shifted pixel stays constant, is unfortunately no longer valid when the brightness in the image sequence changes from one image to the next. A famous example for such a situation is the cloud region of the YOSEMITE sequence. As the gray value constancy assumption does not hold there, methods relying on gray value constancy are not able to estimate the flow field correctly, as for example the results in Fig. 4.18 reveal[7].

In order to deal with non-constant illumination, it is necessary to introduce a constraint that is invariant against brightness changes. One possibility is the assumption that the *gradient* of the shifted pixels stays constant:

$$\nabla I(x+u, y+v, z+1) - \nabla I(x, y, z) = \mathbf{0}. \tag{4.32}$$

Linearization leads to:

$$\begin{aligned} I_{xx}u + I_{xy}v + I_{xz} &= 0 \\ I_{xy}u + I_{yy}v + I_{yz} &= 0 \end{aligned} \tag{4.33}$$

where double subscripts denote second derivatives. Although the gradient can slightly change due to changes in the gray value, too, it is much less dependent on the illumination.

A gradient constancy assumption has been proposed in [UGVT88] and [Tis94] for dealing with the aperture problem in local methods. In the scope of global methods this is not the key issue, as the aperture problem is already addressed by means of the smoothness assumption. Hence a gradient constancy assumption has not yet been applied in global optic flow estimation. It has only been briefly suggested in [Sch94a], yet has not been further investigated. Here in this work, the gradient constancy assumption is elaborately investigated as a cue that is invariant against brightness changes.

[7] Some people claim that there was no ground truth for the cloud region. On his internet page, Michael Black, for instance, writes: *'Some methods report errors on the sequence with clouds and assume that the motion of the clouds is translational. I want to emphasize there is no ground truth for the cloud motion. The cloud pattern is fractal and undergoing Brownian motion. The assumption of brightness constancy does not hold for the clouds. So reporting errors in the cloud region is completely meaningless. For this reason, one should only report errors for the rigid region of the scene.'* `http://www.cs.brown.edu/people/black/Sequences/yosFAQ.html`.
However, there is very well ground truth for the cloud region, which is indeed translational. The problem for most optic flow estimation methods (also the methods of Black) is the fact, that additionally to the translational shift of the clouds the light source behind the clouds increases its intensity over time. This is not Brownian motion but just a local brightness change, which does hardly harm the motion perception of a human observer. With a suitable model that can deal with illumination changes, the correct flow can also be extracted by a machine, as some of the results in Fig. 4.22 show.

However, as the gradient constancy assumption leads to second order derivatives, one can expect a higher sensitivity to noise. Furthermore, the gradient constancy assumption can only provide information at positions in the image where the second derivatives are not zero (in contrast to the brightness constancy assumption, where only the first derivatives must be different from zero). Thus it is proposed to combine both assumptions with a weighting parameter γ:

$$E_{Data} = (I_x u + I_y v + I_z)^2 + \gamma\Big((I_{xx}u + I_{xy}v + I_{xz})^2 + (I_{xy}u + I_{yy}v + I_{yz})^2\Big). \qquad (4.34)$$

This new data term can be joined with a robust function and the smoothness assumptions mentioned above, leading to the energy functional:

$$E(u,v) = \int_\Omega \left(\Psi(E_{Data}) + \alpha\,\Psi(|\nabla u|^2 + |\nabla v|^2)\right)\,\mathbf{dx}. \qquad (4.35)$$

The corresponding Euler-Lagrange equations become a bit longer, yet they are in principle no more complicated than the Euler-Lagrange equations in (4.31):

$$\begin{array}{rcl}
\Psi'(E_{Data})\Big((I_x u + I_y v + I_z)I_x + \gamma\left((I_{xx}u + I_{xy}v + I_{xz})I_{xx} + (I_{xy}u + I_{yy}v + I_{yz})I_{xy}\right)\Big) & & \\
-\alpha\,\mathrm{div}\left(\Psi'(|\nabla u|^2 + |\nabla v|^2)\nabla u\right) & = & 0 \\
\Psi'(E_{Data})\Big((I_x u + I_y v + I_z)I_y + \gamma\left((I_{xx}u + I_{xy}v + I_{xz})I_{xy} + (I_{xy}u + I_{yy}v + I_{yz})I_{yy}\right)\Big) & & \\
-\alpha\,\mathrm{div}\left(\Psi'(|\nabla u|^2 + |\nabla v|^2)\nabla v\right) & = & 0.
\end{array} \qquad (4.36)$$

Numerics. Before considering a numerical scheme for minimizing the energy in (4.35), it is wise to start with the much simpler energy in (4.24). Discretization of the Euler-Lagrange equations leads to a linear system of equations, where for each pixel i there are two rows in the system matrix (one for u_i and one for v_i). The sparse linear system can be solved quite efficiently by the successive over-relaxation (SOR) method [You71]. Starting with an initialization $(\mathbf{u}^0, \mathbf{v}^0) = \mathbf{0}$, the SOR iteration scheme for minimizing the energy in (4.24) reads:

$$\begin{array}{rcl}
u_i^{k+1} & = & (1-\omega)\,u_i^k + \omega\,\dfrac{\sum\limits_{j\in\mathcal{N}^-(i)} u_j^{k+1} + \sum\limits_{j\in\mathcal{N}^+(i)} u_j^k - \dfrac{1}{\alpha}((I_xI_y)_i v_i^k + (I_xI_z)_i)}{\sum\limits_{j\in\mathcal{N}^-(i)\cup\mathcal{N}^+(i)} 1 \qquad + \dfrac{1}{\alpha}(I_x^2)_i} \\
v_i^{k+1} & = & (1-\omega)\,v_i^k + \omega\,\dfrac{\sum\limits_{j\in\mathcal{N}^-(i)} v_j^{k+1} + \sum\limits_{j\in\mathcal{N}^+(i)} v_j^k - \dfrac{1}{\alpha}((I_xI_y)_i u_i^{k+1} + (I_yI_z)_i)}{\sum\limits_{j\in\mathcal{N}^-(i)\cup\mathcal{N}^+(i)} 1 \qquad + \dfrac{1}{\alpha}(I_y^2)_i}
\end{array}$$

where $\mathcal{N}^-(i)$ denotes the neighbors j of i with $j < i$, and $\mathcal{N}^+(i)$ the neighbors j of i with $j > i$. The scheme converges for the relaxation parameter $\omega \in (0,2)$. The choice of ω for which the scheme shows the best convergence depends on the linear system that has to be solved. Like for the linear system in Section 2.3.1, also here values close to 2 show experimentally the best performance. Considerable speedups are possible by employing multi-grid strategies to solve the linear system, as proposed in [BWF+03, BWF+05].

With a non-quadratic penalizer function in either the data or the smoothness term, the Euler-Lagrange equations are no longer linear in u and v. Consequently, a discretization of (4.31) leads to a nonlinear system of equations. As mentioned in Section 2.3.2 the nonlinearity in the

Euler-Lagrange equations can be removed by means of a fixed point iteration scheme. Keeping u and v fixed for computing the expressions $\Psi'((I_x u + I_y v + I_z)^2)$ and $\Psi'(|\nabla u|^2 + |\nabla v|^2)$ yields again linear equations, which can be solved after discretization by means of SOR. So there are two iteration loops: one inner loop that solves the linear systems and one outer loop for the fixed point iterations.

Let $\mathbf{w}^k = (u^k, v^k, 1)$ denote the fixed point after iteration k, and let $\mathbf{w}^0 = (0, 0, 1)$. Then $\mathbf{w}^{k+1}$ is the solution of

$$\begin{aligned}
\Psi'((I_x u^k + I_y v^k + I_z)^2)(I_x u^{k+1} + I_y v^{k+1} + I_z)I_x - \alpha \operatorname{div}\left(\Psi'(|\nabla u^k|^2 + |\nabla v^k|^2)\nabla u^{k+1}\right) &= 0 \\
\Psi'((I_x u^k + I_y v^k + I_z)^2)(I_x u^{k+1} + I_y v^{k+1} + I_z)I_y - \alpha \operatorname{div}\left(\Psi'(|\nabla u^k|^2 + |\nabla v^k|^2)\nabla v^{k+1}\right) &= 0
\end{aligned}$$

where ${\Psi'_D}^k := \Psi'((I_x u^k + I_y v^k + I_z)^2)$ can be interpreted as a robustness factor in the data term, and ${\Psi'_S}^k := \Psi'(|\nabla u^k|^2 + |\nabla v^k|^2)$ as a diffusivity in the smoothness term. Discretization now yields a linear system of equations. Let l denote the iteration index for the SOR iterations, then the iteration scheme for solving the linear system is:

$$\begin{aligned}
u_i^{k,l+1} &= (1-\omega)\, u_i^{k,l} + \omega\, \frac{\displaystyle\sum_{j\in\mathcal{N}^-(i)} (\Psi'_S)^k_{i\sim j} u_j^{k,l+1} + \sum_{j\in\mathcal{N}^+(i)} (\Psi'_S)^k_{i\sim j} u_j^{k,l} - \frac{(\Psi'_D)^k_i}{\alpha}\left((I_x I_y)_i v_i^{k,l} + (I_x I_z)_i\right)}{\displaystyle\sum_{j\in\mathcal{N}^-(i)\cup\mathcal{N}^+(i)} (\Psi'_S)^k_{i\sim j} + \frac{(\Psi'_D)^k_i}{\alpha}(I_x^2)_i} \\
v_i^{k,l+1} &= (1-\omega)\, v_i^{k,l} + \omega\, \frac{\displaystyle\sum_{j\in\mathcal{N}^-(i)} (\Psi'_S)^k_{i\sim j} v_j^{k,l+1} + \sum_{j\in\mathcal{N}^+(i)} (\Psi'_S)^k_{i\sim j} v_j^{k,l} - \frac{(\Psi'_D)^k_i}{\alpha}\left((I_x I_y)_i u_i^{k,l+1} + (I_y I_z)_i\right)}{\displaystyle\sum_{j\in\mathcal{N}^-(i)\cup\mathcal{N}^+(i)} (\Psi'_S)^k_{i\sim j} + \frac{(\Psi'_D)^k_i}{\alpha}(I_y^2)_i}
\end{aligned}$$

where the expression $(\Psi'_S)^k_{i\sim j}$ denotes the discrete diffusivity between pixels i and j. See Section 2.1.3 for discretization details.

In principle, this iteration scheme contains all relevant techniques that are necessary to compute also a minimizer of the energy in (4.35). The corresponding iteration equations to minimize this energy are only considerably larger. For completeness and because the technique in the next section is based on a similar energy functional the equations are given below. With the new abbreviation for the robustness factor

$$(\Psi'_D)^k := \Psi'\Big((I_x u^k + I_y v^k + I_z)^2 + \gamma\left((I_{xx} u^k + I_{xy} v^k + I_{xz})^2 + (I_{xy} u^k + I_{yy} v^k + I_{yz})^2\right)\Big)$$

the fixed point iteration scheme reads

$$\begin{aligned}
&(\Psi'_D)^k\Big((I_x u^{k+1} + I_y v^{k+1} + I_z)I_x + \gamma\left((I_{xx} u^{k+1} + I_{xy} v^{k+1} + I_{xz})I_{xx} + (I_{xy} u^{k+1} + I_{yy} v^{k+1} + I_{yz})I_{xy}\right)\Big) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad -\alpha \operatorname{div}\left((\Psi'_S)^k \nabla u^{k+1}\right) = 0
\end{aligned}$$

$$\begin{aligned}
&(\Psi'_D)^k\Big((I_x u^{k+1} + I_y v^{k+1} + I_z)I_y + \gamma\left((I_{xx} u^{k+1} + I_{xy} v^{k+1} + I_{xz})I_{xy} + (I_{xy} u^{k+1} + I_{yy} v^{k+1} + I_{yz})I_{yy}\right)\Big) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad -\alpha \operatorname{div}\left((\Psi'_S)^k \nabla v^{k+1}\right) = 0
\end{aligned}$$

and the update equations with SOR are:

$$\begin{aligned}
u_i^{k,l+1} &= (1-\omega)\, u_i^{k,l} \\
&+ \omega \frac{\sum\limits_{j\in\mathcal{N}^-(i)} (\Psi'_S)^k_{i\sim j} u_j^{k,l+1} + \sum\limits_{j\in\mathcal{N}^+(i)} (\Psi'_S)^k_{i\sim j} u_j^{k,l}}{\sum\limits_{j\in\mathcal{N}^-(i)\cup\mathcal{N}^+(i)} (\Psi'_S)^k_{i\sim j} + \frac{(\Psi'_D)^k_i}{\alpha}((I_x^2)_i + (I_{xy})^2_i + (I_{xx})^2_i)} \\
&- \omega \frac{\frac{(\Psi'_D)^k_i}{\alpha}\Big((I_xI_y)_i v_i^{k,l} + (I_xI_z)_i + \gamma\Big(((I_{xy})_i v_i^{k,l} + (I_{xz})_i)(I_{xx})_i + ((I_{yy})_i v_i^{k,l} + (I_{yz})_i)(I_{xy})_i\Big)\Big)}{\sum\limits_{j\in\mathcal{N}^-(i)\cup\mathcal{N}^+(i)} (\Psi'_S)^k_{i\sim j} + \frac{(\Psi'_D)^k_i}{\alpha}((I_x^2)_i + (I_{xy})^2_i + (I_{xx})^2_i)}
\end{aligned}$$

$$\begin{aligned}
v_i^{k,l+1} &= (1-\omega)\, v_i^{k,l} \\
&+ \omega \frac{\sum\limits_{j\in\mathcal{N}^-(i)} (\Psi'_S)^k_{i\sim j} v_j^{k,l+1} + \sum\limits_{j\in\mathcal{N}^+(i)} (\Psi'_S)^k_{i\sim j} v_j^{k,l}}{\sum\limits_{j\in\mathcal{N}^-(i)\cup\mathcal{N}^+(i)} (\Psi'_S)^k_{i\sim j} + \frac{(\Psi'_D)^k_i}{\alpha}((I_y^2)_i + (I_{xy})^2_i + (I_{yy})^2_i)} \\
&- \omega \frac{\frac{(\Psi'_D)^k_i}{\alpha}\Big((I_xI_y)_i u_i^{k,l+1} + (I_yI_z)_i + \gamma\Big(((I_{xx})_i u_i^{k,l+1} + (I_{xz})_i)(I_{xy})_i + ((I_{xy})_i u_i^{k,l+1} + (I_{yz})_i)(I_{yy})_i\Big)\Big)}{\sum\limits_{j\in\mathcal{N}^-(i)\cup\mathcal{N}^+(i)} (\Psi'_S)^k_{i\sim j} + \frac{(\Psi'_D)^k_i}{\alpha}((I_y^2)_i + (I_{xy})^2_i + (I_{yy})^2_i)}.
\end{aligned}$$

Experiments. The results of several experiments are shown in Fig. 4.22 - Fig. 4.24 and in Tab. 4.2 in order to analyze the impact of each extension to the basic energy functional, and to demonstrate the superiority of global optic flow estimation when compared to the local methods from the last section.
For all experiments the same robust penalizer function $\Psi(s^2) = \sqrt{s^2+\epsilon^2}$ with $\epsilon = 0.001$ has been used. Also the relaxation parameter ω of the SOR scheme has been kept fixed at $\omega = 1.99$. The number of fixed point iterations and SOR iterations is indicated in Tab. 4.2, where the first of the two values specifies the fixed point iterations. There has been no presmoothing of the image sequences in temporal direction.

Probably the most interesting test sequence here is the YOSEMITE sequence. It contains both divergent and translational motion, one large and several smaller motion discontinuities, and local brightness changes in the sky region. Consequently, one can observe in Fig. 4.22 and Tab. 4.2 that, starting from the classic Horn-Schunck approach, each extension of the optic flow model leads to a significant improvement both in the visual impression and the average angular error (AAE).

First the introduction of a non-quadratic smoothness term allows the model to better capture the motion discontinuities. The extension of this model to the spatio-temporal domain leads to further improvements due to the availability of more data. Applying the robust function also to the data term addresses problems at the boundaries of the image sequence, where occlusions occur and therefore outliers in the data compromise the correct estimation of the flow field. Finally, the introduction of the gradient constancy assumption allows for the correct estimation of the motion even in the sky region, where local brightness changes are present.

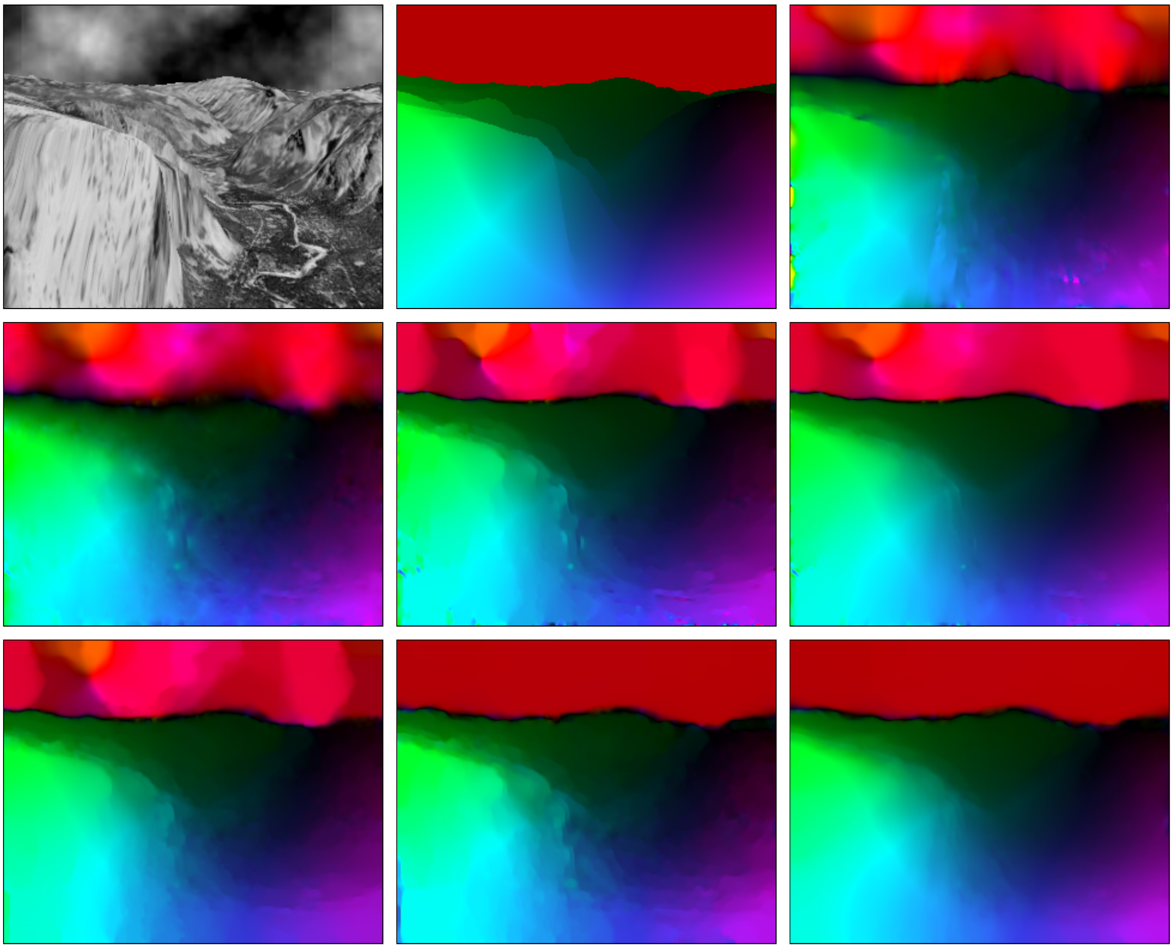


Fig. 4.22: YOSEMITE sequence with cloudy sky ($316 \times 252 \times 15$).
TOP ROW: LEFT: Frame 8. MIDDLE: Correct flow field between frame 8 and 9. RIGHT: Best performing local method (Nonlinear Lucas-Kanade 3-D).
MIDDLE ROW: LEFT: Classic Horn-Schunck. MIDDLE: Non-quadratic smoothness term. RIGHT: Non-quadratic smoothness term 3-D.
BOTTOM ROW: LEFT: Non-quadratic data and smoothness term. MIDDLE: Gradient constancy assumption. RIGHT: Gradient constancy assumption 3-D.

One can also see that the global optic flow estimation methods in general perform better than the local methods. Even though the best performing local method from Section 4.2.1 has been chosen for comparison, already the supplement of a non-quadratic smoothness term to the basic global method leads to more accurate results.

A similar conclusion can be drawn when regarding the results for the STREET sequence and the MARBLE sequence. Also here, each extension of the model entails an improvement in the average angular error and the variational model shows its clear superiority towards local methods.

For the STREET sequence, again the extension of the model by a non-quadratic smoothness term suffices to obtain results that are significantly more accurate than those feasible with the local method. An exception in this regard is the MARBLE sequence. In this sequence, spatio-temporal smoothness is very important. Thus the local method with spatio-temporal flow constancy assumption yields better flow estimates than global methods with a purely spatial smoothness assumption. However, global methods with the smoothness assumption extended to the spatio-temporal domain can again clearly outperform the best local method.

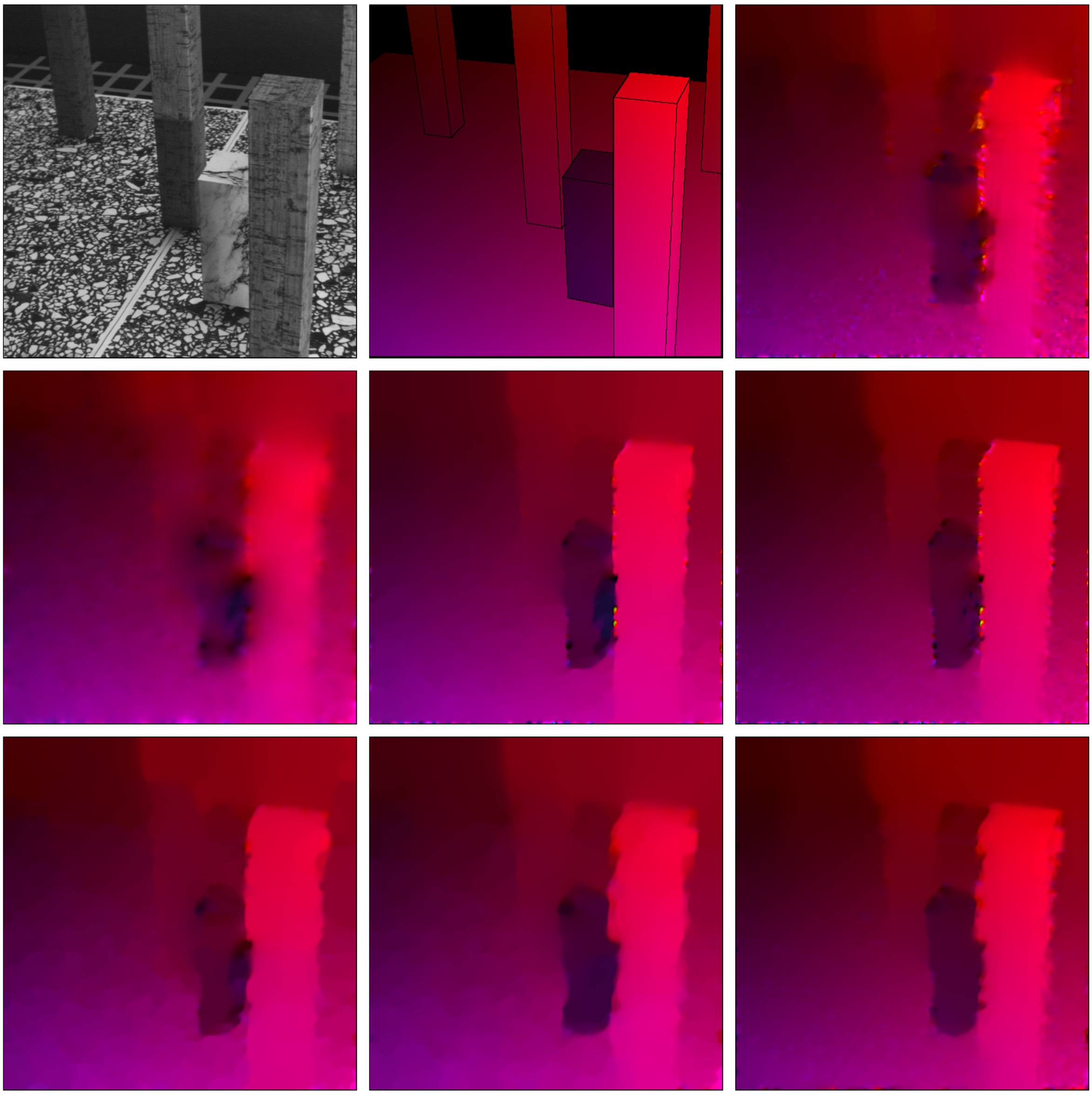


Fig. 4.23: MARBLE sequence ($512 \times 512 \times 31$).
TOP ROW: LEFT: Frame 16. MIDDLE: Correct flow field between frame 16 and 17. RIGHT: Best performing local method (Nonlinear Lucas-Kanade 3-D).
MIDDLE ROW: LEFT: Classic Horn-Schunck. MIDDLE: Non-quadratic smoothness term. RIGHT: Non-quadratic smoothness term 3-D.
BOTTOM ROW: LEFT: Non-quadratic data and smoothness term. MIDDLE: Gradient constancy assumption. RIGHT: Gradient constancy assumption 3-D.

Moreover, note that for the experiment with the local method the amount of pre-smoothing along the temporal axis σ_z has been optimized, while for simplicity this parameter has been set to 0 in the global methods.

Since there are no illumination changes in the STREET sequence, the influence of the gradient constancy assumption has to be reduced considerably in order to yield optimal results. It is interesting, however, that the optimum is not reached for $\gamma = 0$, but for a slightly larger value. This can be explained by the fact that the gradient constancy assumption provides not only illumination invariance but also two additional constraints which help to determine the optic flow.

Yosemite sequence.

Technique	σ	α	γ	Iterations	AAE	Std. dev.
Classic Horn-Schunck	1.3	500	0	500	7.17°	±9.09°
Best performing local method	1.3	-	0	-	6.82°	±10.62°
Non-quadratic smoothness term	1.3	40	0	20/100	6.36°	±8.34°
Non-quadratic data and smoothness term	1.5	25	0	20/100	5.97°	±7.79°
Non-quadratic smoothness term 3-D	1.3	40	0	20/50	5.66°	±7.96°
Gradient constancy assumption	2.0	80	300	20/100	3.50°	±7.84°
Gradient constancy assumption 3-D	2.3	65	400	20/50	2.76°	±7.58°

Street sequence.

Technique	σ	α	γ	Iterations	AAE	Std. dev.
Classic Horn-Schunck	0.5	1100	0	500	10.40°	±15.43°
Best performing local method	0.4	-	0	-	7.75°	±12.50°
Non-quadratic smoothness term	0.3	2000	0	20/100	6.64°	±12.35°
Non-quadratic data and smoothness term	0.6	35	0	20/100	6.31°	±13.43°
Non-quadratic smoothness term 3-D	0.5	450	0	20/50	6.17°	±12.73°
Gradient constancy assumption	0.8	40	0.5	20/100	6.02°	±14.07°
Gradient constancy assumption 3-D	0.9	25	0.25	20/50	5.48°	±15.48°

Marble sequence.

Technique	σ	α	γ	Iterations	AAE	Std. dev.
Non-quadratic smoothness term	2.25	30	0	20/100	5.33°	±3.38°
Classic Horn-Schunck	2.6	1000	0	500	5.31°	±3.48°
Non-quadratic data and smoothness term	3.5	20	0	20/100	5.07°	±3.57°
Gradient constancy assumption	4.5	40	220	20/100	4.83°	±2.99°
Best performing local method	2.5	-	0	-	3.24°	±4.22°
Non-quadratic smoothness term 3-D	2.25	10	0	20/50	2.74°	±3.66°
Gradient constancy assumption 3-D	3.5	20	50	20/50	2.46°	±3.15°

Tab. 4.2: Quantitative evaluation of the methods. AAE = average angular error. The parameter σ denotes the standard deviation of the Gaussian kernel used for pre-smoothing the image sequence in spatial direction.

Since the Marble sequence is not synthetic but has been obtained with a real camera, there are flickering effects visible. Consequently, the gradient constancy assumption is again important for this sequence, though there are not such obvious illumination changes like in the Yosemite sequence.

One can see that quite a few iterations are necessary for the methods to obtain reasonable results. Thus computation times are in many cases above one minute per frame. Apart from code optimization, multi-grid methods, as recently proposed in [BWKS05], are a promising way towards real-time computation. Also multi-resolution strategies can respectably reduce computation times, besides their improved performance. They are the topic of the next section.

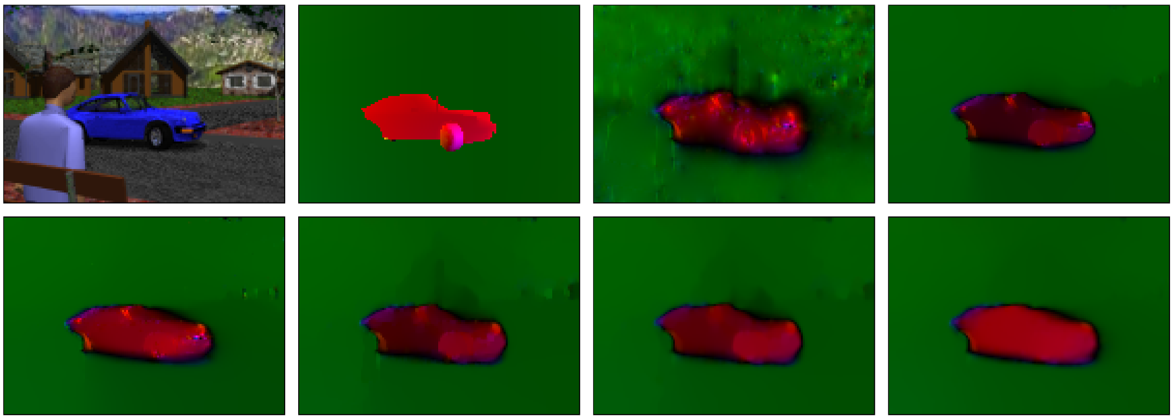


Fig. 4.24: Street sequence ($140 \times 100 \times 20$).

Top Row: From Left to Right: (a) Frame 10. (b) Correct flow field between frame 10 and 11. (c) Classic Horn-Schunck. (d) Non-quadratic smoothness term.

Bottom Row: From Left to Right: (e) Non-quadratic smoothness term 3-D. (f) Non-quadratic data and smoothness term. (g) Gradient constancy assumption. (h) Gradient constancy assumption 3-D.

4.2.3 Method with Non-linearized Constancy Assumptions

At the beginning of this optic flow section, it has been mentioned that linearization of the data constancy assumption

$$I(x, y, z) = I(x + u, y + v, z + 1)$$

to the optic flow constraint equation

$$I_x u + I_y v + I_z = 0$$

is in general a rather bad approximation of the original constraint. In particular in the case of large displacements the image does not change linearly along the displacement vector. Hence, though the linearization is a very convenient way to tackle the actually difficult nonlinear optic flow estimation problem, it is also the reason for relatively unreliable estimation results as soon as the displacements are significantly larger than one pixel per frame.

In this section it is therefore proposed to work with the original non-linearized equation while the linearization is postponed to the numerical scheme. This is rewarded by improved results in all test sequences, especially in the presence of large displacements.

The idea to refrain from a linearization of the data constancy assumption is not completely new. It has been proposed, e.g., in [NE86, AWS00]. However, it is coupled here with a multi-resolution strategy, which not only yields improved results when compared to the method in [AWS00], but also shows strong similarities to so-called *warping methods* like the one proposed in [MP98a, MP98b, MP02].

Variational model. Applying non-quadratic penalizer functions to both the data and the smoothness term and also integrating the gradient constancy assumption as proposed in the last section, the optic flow model is described by the following energy functional:

$$E(u, v) = \int_\Omega \left(\Psi\Big(\big(I(\mathbf{x} + \mathbf{w}) - I(\mathbf{x})\big)^2 + \gamma |\nabla I(\mathbf{x} + \mathbf{w}) - \nabla I(\mathbf{x})|^2 \Big) + \alpha\, \Psi(|\nabla u|^2 + |\nabla v|^2) \right) \mathbf{dx} \quad (4.37)$$

with the optic flow vector $\mathbf{w} := (u, v, 1)$. The only difference to the energy in (4.35) is the missing linearization of the data constraint. However, this seemingly small difference has a large impact on the difficulty of the optimization problem. While for the optimization problem in (4.35) a unique minimum can be guaranteed for convex penalizer functions Ψ, this is certainly no longer the case for the optimization problem stated here. Thus one has to deal with local optima during the optimization process.

Euler-Lagrange equations. A (local) minimum of this energy has to fulfill the Euler-Lagrange equations. For better readability, the following abbreviations are used:

$$\begin{array}{lll} I_x & := & \partial_x I(\mathbf{x}+\mathbf{w}) \\ I_y & := & \partial_y I(\mathbf{x}+\mathbf{w}) \\ I_z & := & I(\mathbf{x}+\mathbf{w}) - I(\mathbf{x}) \\ I_{xx} & := & \partial_{xx} I(\mathbf{x}+\mathbf{w}) \\ I_{xy} & := & \partial_{xy} I(\mathbf{x}+\mathbf{w}) \\ I_{yy} & := & \partial_{yy} I(\mathbf{x}+\mathbf{w}) \\ I_{xz} & := & \partial_x I(\mathbf{x}+\mathbf{w}) - \partial_x I(\mathbf{x}) \\ I_{yz} & := & \partial_y I(\mathbf{x}+\mathbf{w}) - \partial_y I(\mathbf{x}). \end{array} \tag{4.38}$$

Note that expressions containing z are *not* partial derivatives in z, but differences that are sought to be minimized. This is in contrast to the last section.
With the abbreviations it is possible to write the Euler-Lagrange equations in a more compact manner:

$$\begin{array}{lll} \Psi'(I_z^2 + \gamma(I_{xz}^2 + I_{yz}^2)) \cdot (I_x I_z + \gamma(I_{xx} I_{xz} + I_{xy} I_{yz})) - \alpha \text{ div } \left(\Psi'(|\nabla u|^2 + |\nabla v|^2) \nabla u\right) & = & 0 \\ \Psi'(I_z^2 + \gamma(I_{xz}^2 + I_{yz}^2)) \cdot (I_y I_z + \gamma(I_{yy} I_{yz} + I_{xy} I_{xz})) - \alpha \text{ div } \left(\Psi'(|\nabla u|^2 + |\nabla v|^2) \nabla v\right) & = & 0 \end{array} \tag{4.39}$$

with reflecting boundary conditions like in the section before.

One observes immediately that the Euler-Lagrange equations are nonlinear in their argument $\mathbf{w} = (u, v, 1)$. We encountered nonlinear Euler-Lagrange equations already in the last section due to non-quadratic penalizers. This kind of nonlinearity is also present in the equations here. Additionally, the equations contain nonlinearities because of the non-linearized data constraints. Consequently, two different kinds of nonlinearities have to be removed before one obtains linear equations that can be resolved. The proposed numerical scheme performs this linearization by two nested fixed point iterations.

Multi-resolution strategy. Besides additional nonlinearities, there is the supplementary problem of local minima, as a multi-modal functional is sought to be optimized. As stated in Section 2.3.3, there is no efficient method that can guarantee a global optimum for such functionals. However, with the concept of continuation methods or GNC methods [BZ87] one can obtain considerably more satisfactory results than with a conventional optimization strategy, as the problem is first considered on a highly simplified, smoothed version of the functional. The idea is that the result of this optimization is a good initialization for successively refined versions of the problem[8].

A good approximation for smoothing the energy functional is to smooth the underlying images. As the smoothing of the images removes small details that are responsible for local minima, one can expect that the energy functional containing the smoothed images has considerably less local minima. The same strategy has been applied in [AWS00]. A nice theoretical investigation on this topic has been performed in [LC01].

[8] For an illustration of continuation methods see Section 2.3.3.

Instead of costly smoothing operations on the originally sized images, it is also possible to downsample the images respecting the sampling theorem. As this removes small details the same way as a smoothing operation on the original image, but additionally leads to a much more efficient multi-resolution implementation, this procedure is chosen here. In theory, continuation methods perform best, if the transition from coarse to fine is as smooth as possible. Each step in the pyramid can cause the initialization at a finer scale to be too close to a local minimum just appearing at this scale. This risk is reduced by making smaller steps, as one may already be closer to the global optimum when the local minimum appears. Therefore, it is suggested to use a downsampling factor $\eta \in (0,1)$ between successive resolution levels which is larger than the commonly chosen 0.5.

Numerical scheme. The proposed numerical scheme combines the continuation method implemented by a multi-resolution strategy with two nested fixed point iterations to remove the nonlinearities in the equations.

The full pyramid of images is used, starting with the smallest possible image at the coarsest grid. Let k be the iteration index for the outer fixed point iteration loop, which is to remove the nonlinearity resulting from the nonlinear data constraints. The fixed point $\mathbf{w}^k = (u^k, v^k, 1)$ is initialized with $\mathbf{w}^0 = (0,0,1)$ on the coarsest grid. Further let I_*^k denote the abbreviations defined in (4.38) with $\mathbf{w}^k$ in place of $\mathbf{w}$. Then $\mathbf{w}^{k+1}$ is the solution of

$$\begin{aligned}
\Psi'((I_z^{k+1})^2 + \gamma((I_{xz}^{k+1})^2 + (I_{yz}^{k+1})^2)) \cdot (I_x^k I_z^{k+1} + \gamma(I_{xx}^k I_{xz}^{k+1} + I_{xy}^k I_{yz}^{k+1})) & \\
-\alpha \ \mathrm{div}\left(\Psi'(|\nabla u^{k+1}|^2 + |\nabla v^{k+1}|^2)\nabla u^{k+1}\right) &= 0 \\
\Psi'((I_z^{k+1})^2 + \gamma((I_{xz}^{k+1})^2 + (I_{yz}^{k+1})^2)) \cdot (I_y^k I_z^{k+1} + \gamma(I_{yy}^k I_{yz}^{k+1} + I_{xy}^k I_{xz}^{k+1})) & \\
-\alpha \ \mathrm{div}\left(\Psi'(|\nabla u^{k+1}|^2 + |\nabla v^{k+1}|^2)\nabla v^{k+1}\right) &= 0.
\end{aligned}$$

As soon as a fixed point in $\mathbf{w}^k$ is reached, the resolution is changed to the next finer scale and the result is used as initialization for one fixed point iteration on this scale. The upsampling of $\mathbf{w}^k$ is carried out by bilinear interpolation.

Notice that this scheme is fully implicit in the smoothness term and semi-implicit in the data term. Implicit schemes used to yield higher stability and faster convergence. However, the new equations are still nonlinear because of the nonlinear function Ψ' and the symbols I_*^{k+1}. In order to remove the nonlinearity in I_*^{k+1}, first order Taylor expansions are used:

$$\begin{aligned}
I_z^{k+1} &\approx I_z^k + I_x^k du^k + I_y^k dv^k \\
I_{xz}^{k+1} &\approx I_{xz}^k + I_{xx}^k du^k + I_{xy}^k dv^k \\
I_{yz}^{k+1} &\approx I_{yz}^k + I_{xy}^k du^k + I_{yy}^k dv^k
\end{aligned}$$

where $u^{k+1} = u^k + du^k$ and $v^{k+1} = v^k + dv^k$. So the unknowns u^{k+1} and v^{k+1} are split into the solutions of the previous iteration step u^k, v^k and unknown increments du^k, dv^k. Expressions of type $I(\mathbf{x} + \mathbf{w}^k)$ are thereby computed by shifting the second image according to the fixed point $\mathbf{w}^k$ towards the first image. Values between grid points are approximated by bilinear interpolation. If values drop out of the image domain, the data term is set to zero there, so the smoothness assumption has to provide the solution.

Note that a first order Taylor expansion serves also for obtaining linearized constancy assumptions in conventional methods. The important difference is, however, that the linearization is now performed *after* a fixed point has been computed. This fixed point can be used to shift the second image towards the first one. This way, the nonlinear constraint is imitated by a sequence of linear approximations instead of one single linear approximation. Consequently, the approximation is much more accurate.

Introducing an abbreviation for the robustness factor Ψ'_D and the diffusivity Ψ'_S,

$$\begin{aligned}(\Psi')^k_D &:= \Psi'\Big((I^k_z + I^k_x du^k + I^k_y dv^k)^2 + \gamma\big((I^k_{xz} + I^k_{xx} du^k + I^k_{xy} dv^k)^2 + (I^k_{yz} + I^k_{xy} du^k + I^k_{yy} dv^k)^2\big)\Big) \\ (\Psi')^k_S &:= \Psi'(|\nabla(u^k + du^k)|^2 + |\nabla(v^k + dv^k)|^2)\end{aligned}$$

the equations for the outer fixed point iteration can be written as:

$$\begin{aligned}(\Psi')^k_D \cdot \Big(I^k_x\,(I^k_z + I^k_x du^k + I^k_y dv^k)\Big) + \gamma\,(\Psi')^k_D \cdot \Big(I^k_{xx}(I^k_{xz} + I^k_{xx} du^k + I^k_{xy} dv^k) + I^k_{xy}(I^k_{yz} + I^k_{xy} du^k + I^k_{yy} dv^k)\Big) & \\ -\alpha \operatorname{div}\big((\Psi')^k_S \nabla(u^k + du^k)\big) &= 0 \\ (\Psi')^k_D \cdot \Big(I^k_y\,(I^k_z + I^k_x du^k + I^k_y dv^k)\Big) + \gamma\,(\Psi')^k_D \cdot \Big(I^k_{xy}(I^k_{xz} + I^k_{xx} du^k + I^k_{xy} dv^k) + I^k_{yy}(I^k_{yz} + I^k_{xy} du^k + I^k_{yy} dv^k)\Big) & \\ -\alpha \operatorname{div}\big((\Psi')^k_S \nabla(v^k + dv^k)\big) &= 0.\end{aligned} \tag{4.40}$$

These are nonlinear equations in the unknown increments du^k and dv^k. The only remaining nonlinearity is due to Ψ'. It has been shown already in the last section that such a kind of nonlinearity can be removed by fixed point iterations, so a second, inner fixed point iteration loop is introduced.

Let l denote the iteration index for this inner loop. The fixed point variables $du^{k,l}$ and $dv^{k,l}$ are both initialized with 0. Furthermore, let $(\Psi')^{k,l}_D$ and $(\Psi')^{k,l}_S$ denote the robustness factor and the diffusivity computed at a fixed point $(du^{k,l}, dv^{k,l})$. This finally leads to the following *linear* equations:

$$\begin{aligned}(\Psi')^{k,l}_D \cdot \Big(I^k_x\,(I^k_z + I^k_x du^{k,l+1} + I^k_y dv^{k,l+1})\Big) + \gamma\,(\Psi')^{k,l}_D \cdot \Big(I^k_{xx}(I^k_{xz} + I^k_{xx} du^{k,l+1} + I^k_{xy} dv^{k,l+1}) + I^k_{xy}(I^k_{yz} + I^k_{xy} du^{k,l+1} + I^k_{yy} dv^{k,l+1})\Big) & \\ -\alpha \operatorname{div}\Big((\Psi')^{k,l}_S \nabla(u^k + du^{k,l+1})\Big) &= 0 \\ (\Psi')^{k,l}_D \cdot \Big(I^k_y\,(I^k_z + I^k_x du^{k,l+1} + I^k_y dv^{k,l+1})\Big) + \gamma\,(\Psi')^{k,l}_D \cdot \Big(I^k_{xy}(I^k_{xz} + I^k_{xx} du^{k,l+1} + I^k_{xy} dv^{k,l+1}) + I^k_{yy}(I^k_{yz} + I^k_{xy} du^{k,l+1} + I^k_{yy} dv^{k,l+1})\Big) & \\ -\alpha \operatorname{div}\Big((\Psi')^{k,l}_S \nabla(v^k + dv^{k,l+1})\Big) &= 0.\end{aligned}$$

Discretization yields a linear system of equations, which can be solved by SOR. Let m denote the iteration index for the SOR iterations, then the iteration scheme for solving the linear system is:

$$\begin{aligned}du_i^{k,l,m+1} &= (1-\omega)\, du_i^{k,l,m} \\ &+ \omega\, \frac{\sum\limits_{j\in\mathcal{N}^-(i)} (\Psi'_S)^{k,l}_{i\sim j}\left(u^k_j + du_j^{k,l,m+1}\right) + \sum\limits_{j\in\mathcal{N}^+(i)} (\Psi'_S)^{k,l}_{i\sim j}\left(u^k_j + du_j^{k,l,m}\right) - \sum\limits_{j\in\mathcal{N}^-(i)\cup\mathcal{N}^+(i)} (\Psi'_S)^{k,l}_{i\sim j} u^k_i}{\sum\limits_{j\in\mathcal{N}^-(i)\cup\mathcal{N}^+(i)} (\Psi'_S)^{k,l}_{i\sim j} + \frac{(\Psi'_D)^{k,l}_i}{\alpha}\left(\left((I_x)^k_i\right)^2 + \left((I_{xy})^k_i\right)^2 + \left((I_{xx})^k_i\right)^2\right)} \\ &- \omega\, \frac{\frac{(\Psi'_D)^{k,l}_i}{\alpha}\left((I_x)^k_i\left((I_y)^k_i dv_i^{k,l,m} + (I_z)^k_i\right) + \gamma\left((I_{xx})^k_i\left((I_{xy})^k_i dv_i^{k,l,m} + (I_{xz})^k_i\right) + (I_{xy})^k_i\left((I_{yy})^k_i dv_i^{k,l,m} + (I_{yz})^k_i\right)\right)\right)}{\sum\limits_{j\in\mathcal{N}^-(i)\cup\mathcal{N}^+(i)} (\Psi'_S)^{k,l}_{i\sim j} + \frac{(\Psi'_D)^{k,l}_i}{\alpha}\left(\left((I_x)^k_i\right)^2 + \left((I_{xy})^k_i\right)^2 + \left((I_{xx})^k_i\right)^2\right)} \\ dv_i^{k,l,m+1} &= (1-\omega)\, dv_i^{k,l,m} \\ &+ \omega\, \frac{\sum\limits_{j\in\mathcal{N}^-(i)} (\Psi'_S)^{k,l}_{i\sim j}\left(v^k_j + dv_j^{k,l,m+1}\right) + \sum\limits_{j\in\mathcal{N}^+(i)} (\Psi'_S)^{k,l}_{i\sim j}\left(v^k_j + dv_j^{k,l,m}\right) - \sum\limits_{j\in\mathcal{N}^-(i)\cup\mathcal{N}^+(i)} (\Psi'_S)^{k,l}_{i\sim j} v^k_i}{\sum\limits_{j\in\mathcal{N}^-(i)\cup\mathcal{N}^+(i)} (\Psi'_S)^{k,l}_{i\sim j} + \frac{(\Psi'_D)^{k,l}_i}{\alpha}\left(\left((I_y)^k_i\right)^2 + \left((I_{xy})^k_i\right)^2 + \left((I_{yy})^k_i\right)^2\right)} \\ &- \omega\, \frac{\frac{(\Psi'_D)^{k,l}_i}{\alpha}\left((I_y)^k_i\left((I_x)^k_i du_i^{k,l,m+1} + (I_z)^k_i\right) + \gamma\left((I_{xy})^k_i\left((I_{xx})^k_i du_i^{k,l,m+1} + (I_{xz})^k_i\right) + (I_{yy})^k_i\left((I_{xy})^k_i du_i^{k,l,m+1} + (I_{yz})^k_i\right)\right)\right)}{\sum\limits_{j\in\mathcal{N}^-(i)\cup\mathcal{N}^+(i)} (\Psi'_S)^{k,l}_{i\sim j} + \frac{(\Psi'_D)^{k,l}_i}{\alpha}\left(\left((I_y)^k_i\right)^2 + \left((I_{xy})^k_i\right)^2 + \left((I_{yy})^k_i\right)^2\right)}\end{aligned}$$

where like in the previous section $(\Psi'_S)^{k,l}_{i\sim j}$ denotes the diffusivity between pixels i and j.

One detail in this numerical scheme should be observed: the multi-resolution strategy implies a rescaling of the derivative filters. As the same discrete filter masks are used for all levels, the relative scale of the filters grows with the decreasing size of the images. As an alternative, one could scale down the filter masks at coarser levels, such that the filter scale is always the same as on the finest grid. This alternative is deliberately set aside in this work. Applying the same filter masks on all resolution levels is motivated by the possibility to interpret the computations on the downsampled image sequences as independent optic flow estimation tasks using the same parameter setting as on the finest grid. While the coarse scale captures the few low frequency correspondences between the image structures still present in the downsampled images, the refined versions consider the (local) high frequency correspondences. The alternative approach would abandon from this interpretation, as it weighs the structures at the varying levels differently. On the other hand, it can be better motivated from the numerical point of view. The details about how to proceed with the derivatives have to be clarified in future research. The way to downscale the filter masks has been persecuted in [Bru05]. The work provides also experimental results.

Relation to so-called 'warping' methods. Coarse-to-fine warping techniques are a frequently used tool for improving the performance of optic flow methods [Ana89, BA96, MP98b, MP02]. While they are often introduced on a purely algorithmic basis, it is shown here that they can be theoretically justified as a numerical approximation of an energy minimization problem based on non-linearized constancy assumptions.

In order to establish this relation, the model in (4.37) is reduced to the gray value constancy assumption by setting $\gamma = 0$. Further on, solely spatial smoothness, as in [MP02], is assumed. Under these conditions, (4.40) simplifies to

$$\begin{aligned}
(\Psi')^k_D \cdot \Big(I^k_x \left(I^k_z + I^k_x du^k + I^k_y dv^k \right) \Big) - \alpha \text{ div } \left((\Psi')^k_S \nabla (u^k + du^k) \right) &= 0 \\
(\Psi')^k_D \cdot \Big(I^k_y \left(I^k_z + I^k_x du^k + I^k_y dv^k \right) \Big) - \alpha \text{ div } \left((\Psi')^k_S \nabla (v^k + dv^k) \right) &= 0.
\end{aligned}$$

For a fixed k, this system is equivalent to the Euler-Lagrange equations described in [MP02]. Also there, the estimation is reduced to the successive computation of increments du and dv between the first image and the warped second image. The same increments appear in the outer fixed point iterations in (4.37). Here they have been introduced in order to resolve the nonlinearity of the gray value constancy assumption. This shows that the warping technique implements the minimization of a non-linearized constancy assumption by means of fixed point iterations on $\mathbf{w}$.

In earlier approaches, the main motivation for warping has been the coarse-to-fine strategy. Due to solutions u and v computed on coarser grids, only an increment du and dv remains to be determined on the fine grid. Thus the estimates used to have a magnitude of less than one pixel per frame, independent from the magnitude of the total displacement. This ability to deal with larger displacements has proven to be a very important aspect in differential optical flow estimation.

The second strategy to deal with large displacements is the utilization of the non-linearized gray value constancy assumption [NE86, AWS00]. In these approaches, large displacements are allowed from the beginning. However, the nonlinearity results in a multi-modal functional. In such a setting, the coarse-to-fine strategy is not necessarily wanted, but inevitable to get closer to the global minimum.

In the end, both strategies not only lead to similar results. In fact, as it is demonstrated

YOSEMITE sequence.

Technique	σ	α	γ	Iterations	AAE	Std. dev.
Linearized constancy assumptions	2.0	80	300	20/100	3.50°	±7.84°
Linearized constancy assumptions 3-D	2.3	65	400	20/50	2.76°	±7.58°
Non-linearized constancy assumptions	1.0	80	100	10/5	2.44°	±6.90°
Non-linearized constancy assumptions 3-D	1.0	60	100	10/5	1.78°	±7.00°

STREET sequence.

Technique	σ	α	γ	Iterations	AAE	Std. dev.
Linearized constancy assumptions	0.8	40	0.5	20/100	6.02°	±14.07°
Linearized constancy assumptions 3-D	0.9	25	0.25	20/50	5.48°	±15.48°
Non-linearized constancy assumptions	0.1	100	6	20/20	4.93°	±13.37°
Non-linearized constancy assumptions 3-D	0.1	50	4	20/20	4.54°	±14.69°

MARBLE sequence.

Technique	σ	α	γ	Iterations	AAE	Std. dev.
Linearized constancy assumptions	4.5	40	220	20/100	4.83°	±2.99°
Non-linearized constancy assumptions	3.75	30	50	10/5	4.73°	±2.81°
Linearized constancy assumptions 3-D	3.5	20	50	20/50	2.46°	±3.15°
Non-linearized constancy assumptions 3-D	1.6	30	30	10/5	1.91°	±2.71°

Tab. 4.3: Comparison of the method with non-linearized constancy assumptions to the linearized version. AAE = average angular error. The parameter σ denotes the standard deviation of the Gaussian kernel used for pre-smoothing the image sequence in spatial direction.

above, they are completely equivalent. As a consequence, the coarse-to-fine warping technique can be formulated as a single minimization problem. On the other hand, image registration techniques, which often rely on non-linearized constancy assumptions, get access to an efficient multi-resolution method for minimizing their complex energy functionals.

Experiments with synthetic sequences. In order to allow for a comparison to optic flow methods with linearized constancy assumptions, the same experiments as in the last section have been performed also with the improved method.
Like before, the robust penalizer function $\Psi(s^2) = \sqrt{s^2 + \epsilon^2}$ with $\epsilon = 0.001$ has been employed for both the data and the smoothness term. Considering the choice of the downsampling factor η, a value of $\eta = 0.95$ has ensured smooth transitions from one scale to the next. Although this rather large value entails also quite a large amount of outer fixed point iterations, it can lead on the other hand to a smaller amount of inner fixed point iterations and SOR iterations necessary to obtain results close to the optimum. Moreover, since most of the outer fixed point iterations are performed on a considerably smaller grid, computation times are often significantly smaller than with the methods presented before. The exact number of iterations is listed in Tab. 4.3, where the first value determines the inner fixed point iterations and the second value the SOR iterations. The number of outer fixed point iterations is determined by the downsampling factor η and therefore only depends on the image size. The SOR parameter has been set to $\omega = 1.8$ in all experiments.

The visual impression of the results depicted in Fig. 4.25 - Fig 4.27 reveal in all test sequences a noticeable improvement when using non-linearized constancy assumptions. This impression is confirmed by the corresponding average angular errors listed in Tab. 4.3.

The method has also been run on the YOSEMITE sequence without clouds[9]. In this test

Yosemite with clouds		
Technique	AAE	STD
Anandan [BFB94]	13.36°	15.64°
Nagel [BFB94]	10.22°	16.51°
Horn–Schunck, mod. [BFB94]	9.78°	16.19°
Uras et al. [BFB94]	8.94°	15.61°
Alvarez et al. [AWS00]	5.53°	7.40°
Mémin–Pérez [MP98b]	4.69°	6.89°
Bruhn et al. [BWS05]	4.17°	7.72°
Proposed method	**2.44°**	**6.90°**
Proposed method 3-D	**1.78°**	**7.00°**

Yosemite without clouds		
Technique	AAE	STD
Black–Jepson [BJ96]	2.29°	2.25°
Ju *et al.* [JBJ96]	2.16°	2.00°
Bab-Hadiashar-Suter [BHS98]	2.05°	2.92°
Lai-Vemuri [LV98]	1.99°	1.41°
Proposed method	**1.65°**	**1.43°**
Mémin-Pérez [MP02]	1.58°	1.21°
Farnebäck [Far01]	1.14°	2.14°
Bruhn et al. [BWS05]	1.02°	1.12°
Proposed method 3-D	**0.997°**	**1.17°**

Tab. 4.4: Comparison of the method with methods from the literature with 100 % density for the Yosemite sequence with and without cloudy sky. AAE = average angular error. STD = standard deviation.

sequence, the cloud region has been replaced by black pixels and the average angular error is not measured there. Thus there are no sharp discontinuities in the motion field anymore, and also the local illumination changes have vanished. Nevertheless, this sequence has been very popular in the literature in recent years and hence presents itself for a comparison of the proposed technique to other state-of-the-art methods.

In this comparison, the quality of the introduced method becomes explicit. For the Yosemite sequence with clouds the technique performs twice as good as the currently best method known from the literature. For the Yosemite sequence without clouds, angular errors below one degree are reached for the first time while providing full density.

The experiments provide also an explanation for this good performance: it lies in the model assumptions clearly stated in the energy functional. Obviously, each extension of the model is important to capture the motion present in a sequence with quite general motion such as the Yosemite sequence: robust penalizer functions applied to the smoothness as well as the data term in order to cope with discontinuities in the motion field and outliers due to occlusions, spatio-temporal smoothness for acquiring additional information, gradient constancy in order to deal with illumination changes, and finally non-linearized constancy constraints to deal with large displacements.

Noise. A comparison to the literature, however, elucidates only some aspects of the quality of the approach. The usage of constancy assumptions that are based on higher order derivatives arises, for example, the question on the robustness under noise. Tab. 4.5 therefore shows the performance of the method when considerable amounts of Gaussian noise with zero mean are added to the sequence[10]. Fig. 4.29 depicts the Yosemite sequence disturbed by the different amounts of noise used for the experiment. It turns out that the method is quite robust under noise despite the utilization of higher order derivatives. The number of iterations has been kept fixed and is equal to the values given for the sequence without noise in Tab. 4.3.

[9] The sequence is available at `http://www.cs.brown.edu/people/black/images.html`.

[10] The random number generator has been initialized with seed 0.

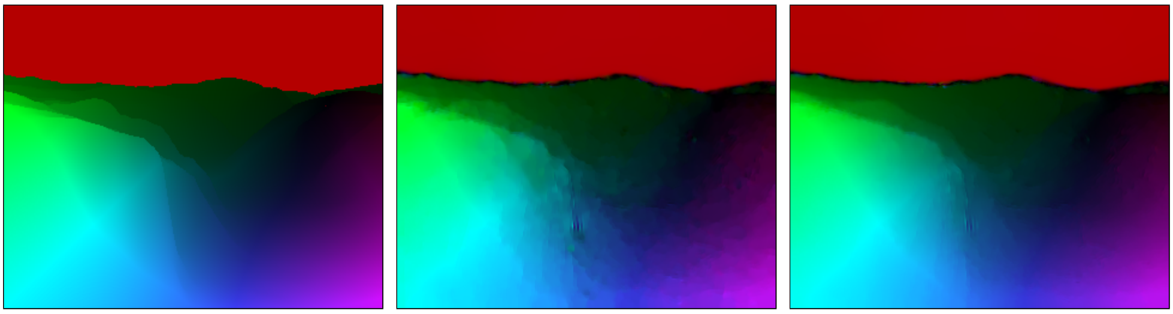


Fig. 4.25: Yosemite sequence with cloudy sky ($316 \times 252 \times 15$) as depicted in Fig. 4.22. Left: Correct flow field. Middle: Non-linearized constancy assumptions. Right: Same method 3-D.

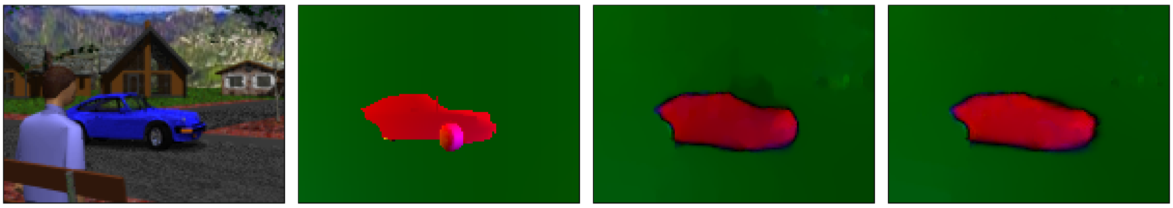


Fig. 4.26: Street sequence ($140 \times 100 \times 20$). From Left to Right: (a) Frame 10. (b) Correct flow field. (c) Non-linearized constancy assumptions. (d) Same method 3-D.

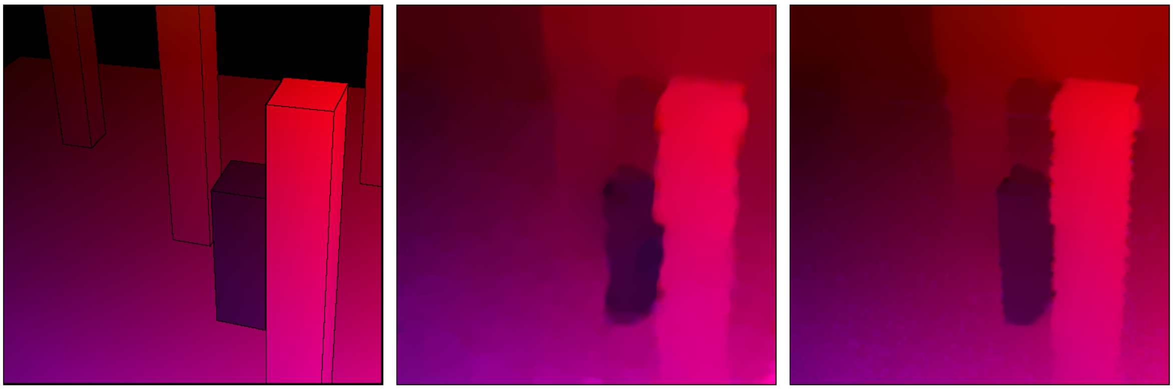


Fig. 4.27: Marble sequence ($512 \times 512 \times 31$) as depicted in Fig. 4.23. Left: Correct flow field. Middle: Non-linearized constancy assumptions. Right: Same method 3-D.

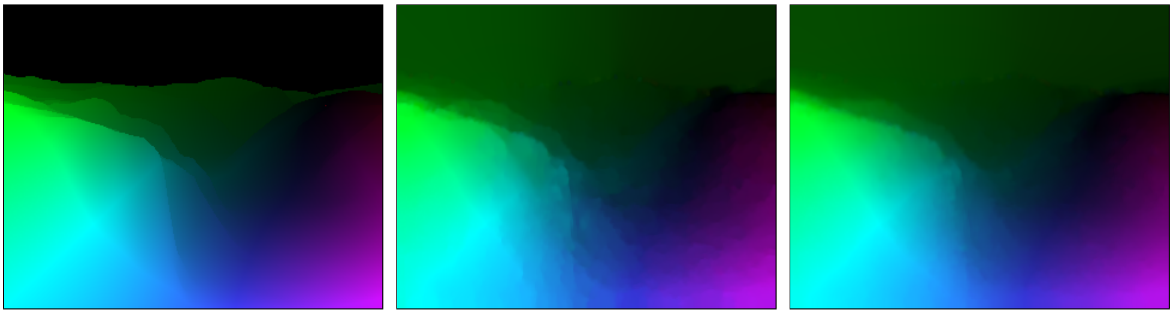


Fig. 4.28: Yosemite sequence with cloud region replaced by black pixels. Left: Correct flow field. Middle: Non-linearized constancy assumptions. Right: Same method 3-D.

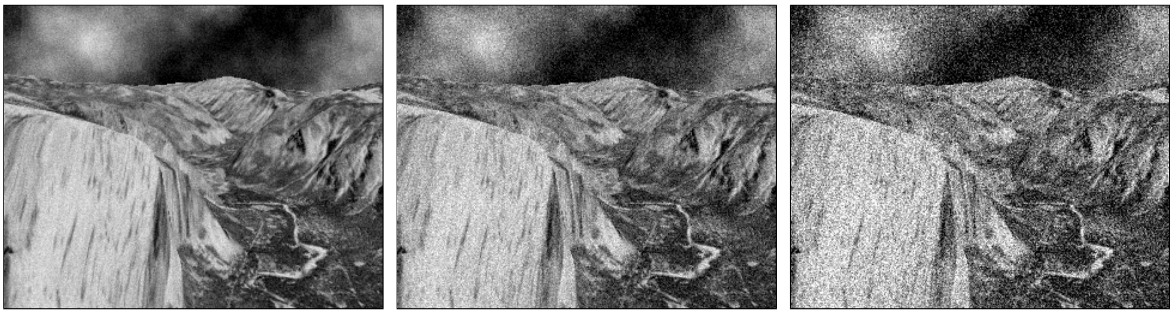


Fig. 4.29: Yosemite sequence with Gaussian noise of standard deviation 10, 20, and 40 added.

σ_n	σ	α	γ	AAE	STD
0	1.0	80	100	1.78°	±7.00°
10	1.6	30	80	2.17°	±6.42°
20	1.8	30	110	2.69°	±6.50°
40	1.9	25	50	4.05°	±7.37°

Tab. 4.5: Noise robustness of the method with spatio-temporal smoothness. Gaussian noise with standard deviation σ_n has been added to the Yosemite sequence with clouds. The parameter σ denotes the standard deviation of the Gaussian kernel used for pre-smoothing the image sequence in spatial direction. AAE = average angular error. STD = standard deviation.

Parameter variation. A further important experiment validates the robustness of the method under parameter variations. Although the habit to optimize the relevant parameters specifically for each image sequence is very useful for optic flow research in order to evaluate and compare different techniques, optic flow applications may often demand for a fixed set of parameters that must work with several different sequences. In Tab. 4.6 the results obtained with such a fixed set of parameters is therefore compared to the optimum results. Although some accuracy is certainly lost because of the suboptimal parameter setting, the quality is still better than with many other methods using optimized parameters. Note that in this test, ideas like adapting the fixed parameter set to the size of the images have not been implemented yet. Such a procedure might push the values much closer to the optimum setting. Other more elaborated approaches for automatic parameter selection are conceivable as well.
Tab. 4.7 shows the impact of parameters deviating by a factor 2 in both directions from the optimum setting. This further suggests that an automatic parameter selection procedure, even if it is not perfect, may lead to very good results.

Sequence	AAE with opt. parameters	AAE with fixed parameters
Yosemite with clouds	2.44°	2.91°
Street	4.93°	6.38°
Marble	4.73°	5.11°
Yosemite without clouds	1.65°	1.76°

Tab. 4.6: Results of the spatial method with the fixed parameter set $\sigma = 1$, $\alpha = 40$, and $\gamma = 20$ using 10 inner fixed point iterations and 5 SOR iterations, in comparison to the results obtained with optimized parameters. AAE = average angular error.

γ_2	σ	α	AAE	STD
100	1.0	60	1.78°	7.00°
50	1.0	60	1.82°	6.89°
200	1.0	60	1.90°	7.23°
100	1.0	30	2.57°	8.40°
100	1.0	120	2.09°	7.48°
100	0.5	60	2.21°	7.10°
100	2.0	60	2.23°	6.85°

Tab. 4.7: Parameter variation in the spatio-temporal method and for the YOSEMITE sequence with clouds. AAE = average angular error. STD = standard deviation.

Computation time. Considering computation times, Tab 4.8 reveals what happens if iterations are stopped long before full convergence has been reached. Obviously, very good results can already be achieved with a fractional amount of computational effort. This way, computation times close to real-time become possible. However, one has to note that other sequences might need more iterations for providing good results. The STREET sequence is such an example, where quite many iterations are necessary. In contrast to the simple reduction of the number of iterations at the expense of quality, the utilization of fast multi-grid techniques is a much more profound way towards a real-time implementation of the method. The interested reader is referred to [BWF$^+$03, BWF$^+$05, BWKS05, Bru05] and the references therein.

Method with spatial smoothness

reduction factor η	outer fixed point iter.	inner fixed point iter.	SOR iter.	computation time	AAE
0.95	77	10	5	14s	2.44°
0.90	38	2	5	1.9s	2.46°
0.85	25	2	5	1.2s	2.63°
0.80	18	2	5	.93s	2.83°
0.75	14	2	5	.78s	3.09°
0.70	12	2	5	.66s	3.38°
0.65	10	2	5	.58s	3.82°
0.60	8	2	3	.45s	4.78°

Method with spatio-temporal smoothness

reduction factor η	outer fixed point iter.	inner fixed point iter.	SOR iter.	computation time/frame	AAE
0.95	77	10	5	19s	1.78°
0.95	77	2	5	4.5s	1.82°
0.90	38	2	5	2.3s	1.98°
0.85	25	2	5	1.6s	2.13°
0.80	18	2	5	1.2s	2.49°
0.75	14	2	5	1.0s	3.20°

Tab. 4.8: Computation times for the Yosemite sequence with clouds.
Computations have been performed on an Intel Pentium 4 processor with 3.06GHz executing compiled C++ code.

Experiments with real-world sequences. Synthetic test sequences are perfect for developing optic flow models and to check whether the model assumptions are appropriate to deal with a specific type of motion or disturbance in the sequence. In the end, however, the methods must be able to handle sequences from real world. Most of the challenges present in the synthetic test sequences, such as noise, illumination changes, or large displacements, also appear here together with additional difficulties. In the following, the method which has proven to meet all requirements for dealing with popular synthetic test sequences is applied to several real-world sequences in order to find its limitations.

Fig. 4.30 - Fig. 4.32 show evaluations in a desk environment. All images used in these experiments have a size of 400×300 pixels. The parameters of the optic flow methods have been kept fixed: there has been no pre-smoothing of the image sequence, $\alpha = 50$ and $\gamma = 10$. The numerical parameters have been $\eta = 0.95$, $\omega = 1.8$, and 10 inner fixed point iterations as well as 5 SOR iterations have been performed.

Translational motion. In the test scenario depicted in Fig. 4.30, the object is shifted to the left, while the camera remains static. This implies actually a very simple translational motion, yet there are several hitches in the scene:

- There is some camera noise, yet it is not expected to cause many problems for the method.
- Already a bit more disturbing is the flickering light source. It would certainly lead to bad estimates using a method without an illumination invariant constancy assumption.
- An artifact not present in synthetic sequences is caused by the auto-focus of the camera. This auto-focus yields different amounts of blurring in the images depending on the object's position.
- There is significantly less texture in the images than in the synthetic sequences. Hence the images provide less information to find pixel correspondences.
- Finally, the increasingly large displacements bring on considerable occlusion problems. Note that only the pixels of the moving object can be retrieved in the second image, yet there is no cue about what has happened to the pixels occluded by the moving object.

The object has been 30cm away from the camera and has been shifted parallel to the image plane by 1cm, 2cm, 4cm, and 8cm. This corresponds to a displacement of the object by around 13.5, 27, 54, and 108 pixels in the image. One can see that the shift by 1cm has been captured very accurately, though the displacement is already quite large compared to displacements in common synthetic test sequences.
In case of the shift by 2cm, occlusion already evokes some problems in the area left to the object. Since there is no information about the displacement of these pixels available in the second image, the smoothness assumption is fully responsible for the result in this area. Note that only the utilization of a non-quadratic penalizer in the data term can ensure for the bad data having that little influence on the result in occluded areas.
The occlusion artifacts gain some more influence when the object is shifted by 4cm. However, the displacement field in the non-occluded areas is still estimated almost correctly.
Only when the object is shifted by 8cm, one can observe that the estimation fails completely. Obviously this happens as soon as the displacement is significantly larger than the object itself. This observation may be explained by the fact that in such situations the object disappears in the coarsened images before the displacement is small enough to capture the object's motion.
In conclusion, the experiment reveals that the method can deal with extraordinary large displacements, yet it has also its limits that are determined by the size of the moving object.

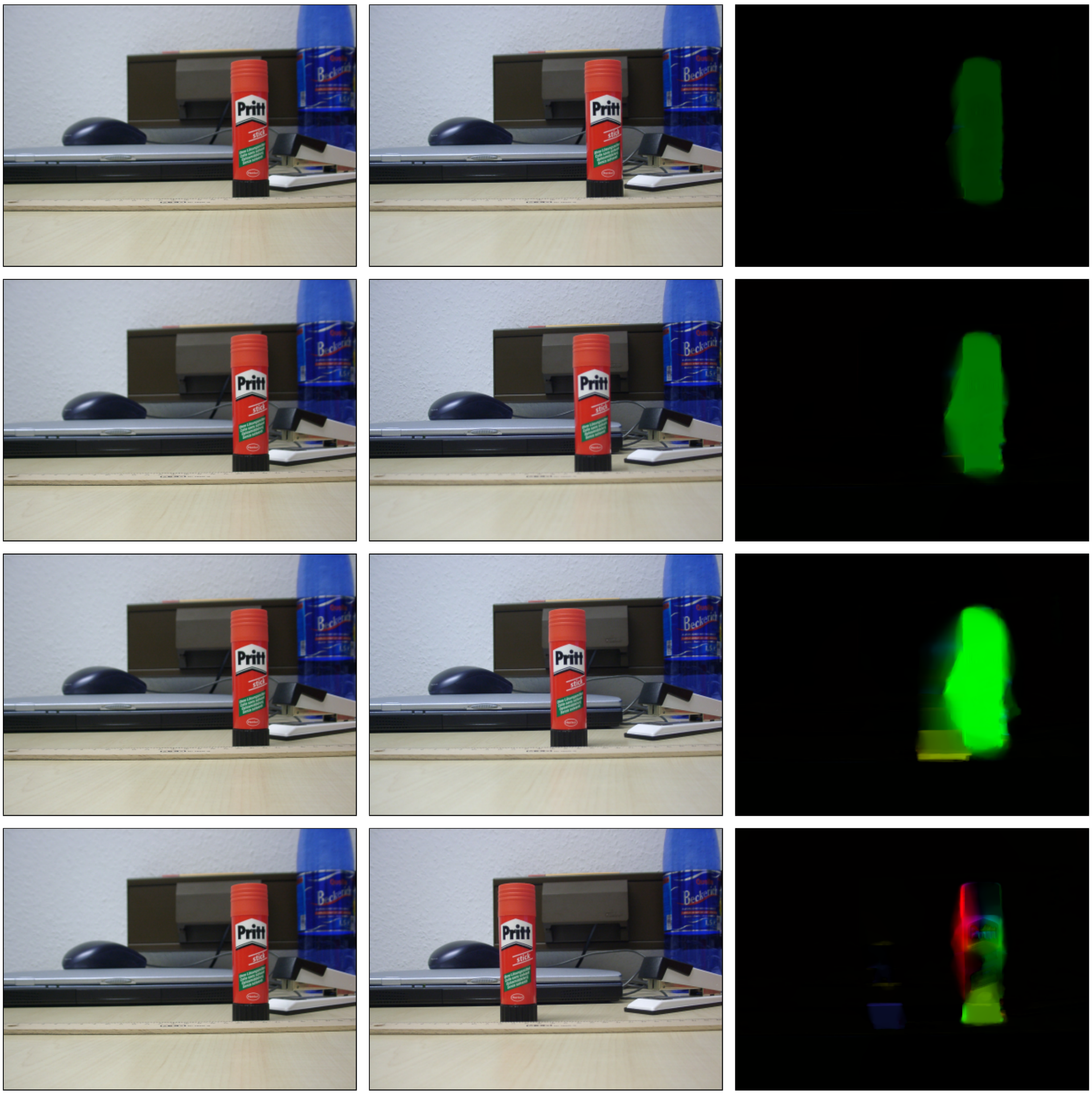


Fig. 4.30: Translating Pritt$^{\text{TM}}$ sequence. The object is 30cm away from the camera and moves to the left. From Left to Right: First image. Second Image. Computed displacement field.
First Row: Object moves 1cm. Computed displacement: around 13 pixels.
Second Row: Object moves 2cm. Computed displacement: around 27 pixels.
Third Row: Object moves 4cm. Computed displacement: around 53 pixels.
Fourth Row: Object moves 8cm. Optic flow estimation failed.

Divergent motion. In the scene depicted in Fig. 4.31, the objects are static while the camera is moving towards them. Similar artifacts as in the experiment before are present in the images. One can observe in particular that the background gets out of focus when the camera moves closer to the foreground object.

Additionally, a more complex motion model, namely divergent motion, comes into play. This kind of motion is well suited for an application called *structure from motion*: projections of 3-D points with a small distance to the camera diverge faster than projections of points that are far away from the camera. Thus one could reconstruct the depth of object points from the optic flow field provided that the camera motion and the camera's intrinsic parameters are known.

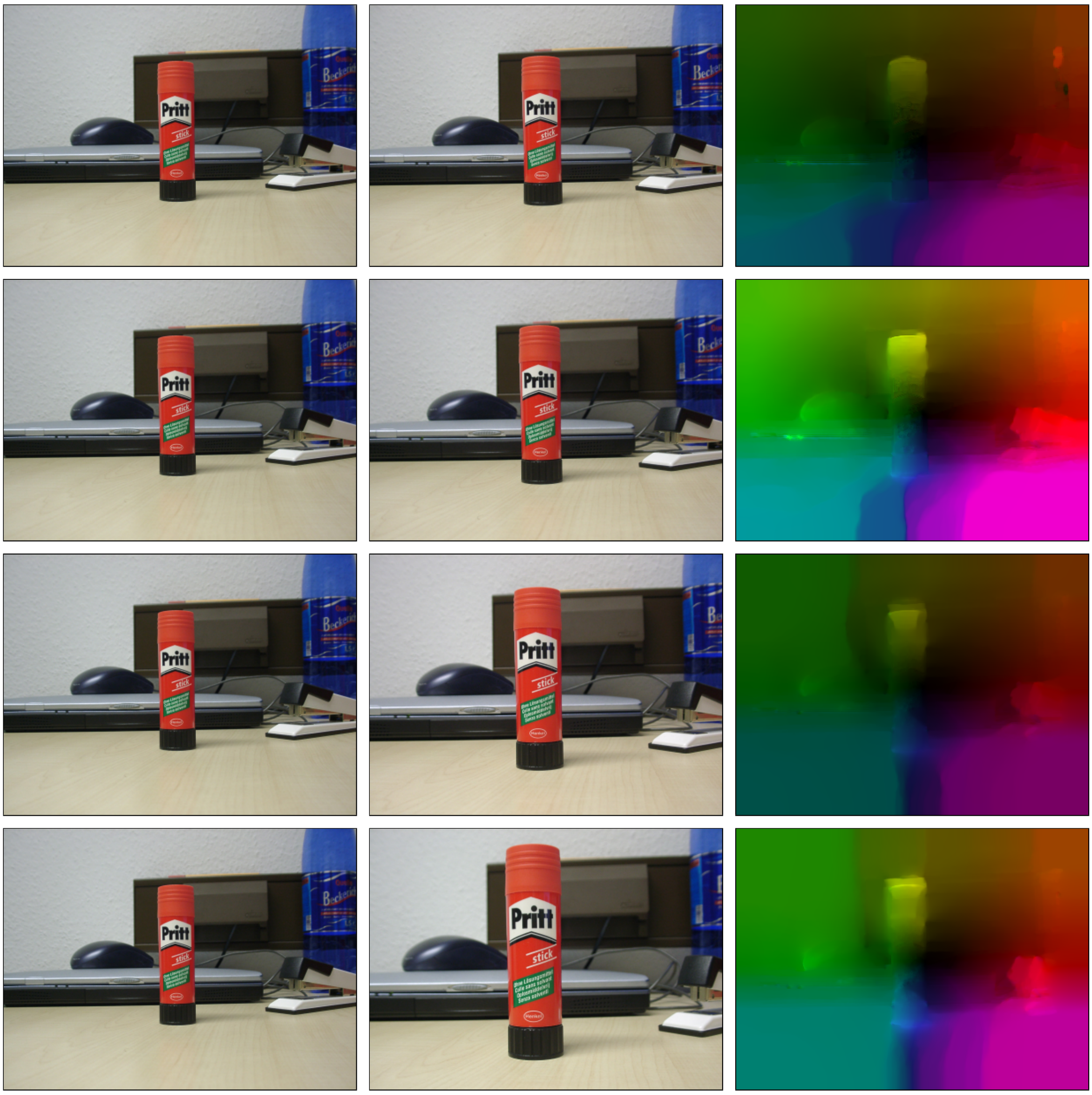


Fig. 4.31: DIVERGING PRITT$^{\mathrm{TM}}$ sequence. The camera is 30cm away from the object and moves towards the object. FROM LEFT TO RIGHT: First image. Second Image. Computed displacement field. FIRST ROW: Camera moves 2cm. SECOND ROW: Camera moves 4cm. THIRD ROW: Camera moves 8cm. FOURTH ROW: Camera moves 12cm. For better visibility, the intensity values in the flow field have been rescaled and are not comparable.

In the first image, the camera has been 30cm away from the central object. Then it has been moved along a straight line towards the object. The camera shift has been 2, 4, 8, and 12cm. The optic flow shows in each case the typical result of a divergent flow field. Moreover, one can see the object boundaries of some foreground objects, especially the stick, the mouse, and the stapler. So indeed pixels belonging to these objects reveal a larger displacement than pixels of the background due to their different depths.

Problems occur in areas with little structure. In such areas, the smoothness term of the optic flow method seems to introduce discontinuities arbitrarily, as the data term provides too little information for more accurate estimates. The problem gains importance with increasingly large displacements. Again this may be explained by the size of available structures relative

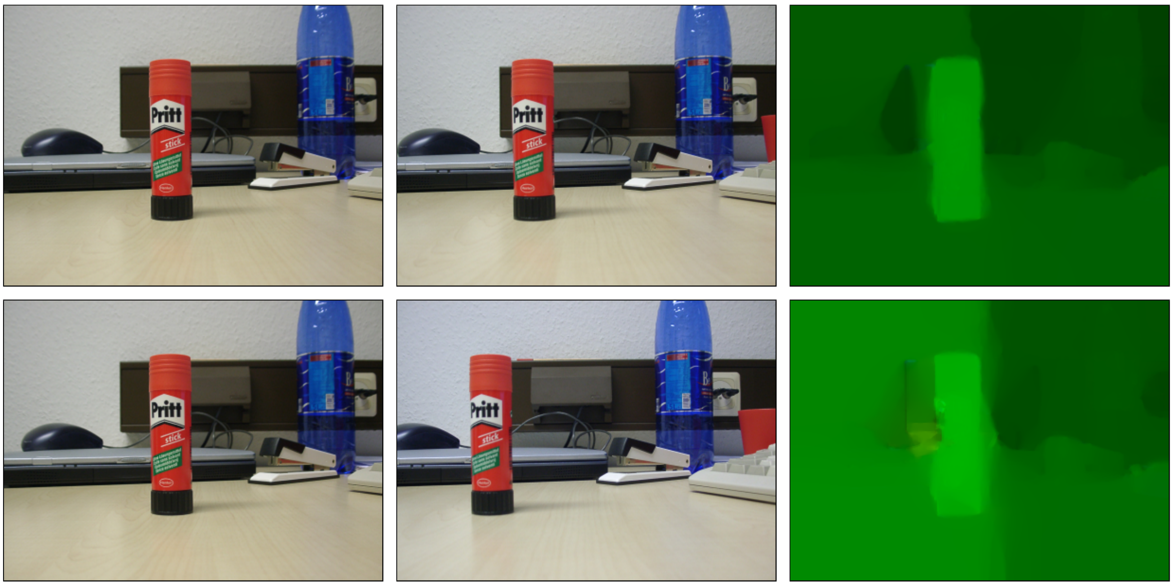


Fig. 4.32: Simulation of a stereo camera system. Camera moves to the right.
From Left to Right: First image. Second Image. Computed displacement field.
First Row: Camera moves 2cm. Second Row: Camera moves 5cm.
For better visibility, the intensity values in the flow field have been rescaled and are not comparable.

to the size of the displacements: small scale structures like the texture of the wallpaper disappear at coarser scales of the images where the cues for large displacements have to be determined. On finer scales, the large displacement impedes any unique correspondence between such small and frequently appearing structures. Consequently, results get worse with increasing displacements, simply because the amount of utilizable information decreases.

On the other hand it is impressive that even in the test case with the largest displacement, the boundaries of objects from the first frame are visible in the optic flow field, although the objects are much larger in the second frame. This shows that the algorithm handles occlusion problems already quite well. Further improvements may be possible with techniques that deal with occlusions explicitly in the model [ADPS02].

Stereo scenario. The last experiment in this series simulates a stereo camera system: in the scenario shown in Fig. 4.32, the camera is shifted in parallel to the image plane by 2cm and 5cm. It is obvious that a larger shift increases the magnitude of the displacement vector. In the scope of stereo vision, the displacement is called *disparity*. It can be used to reconstruct the depth of a point in the image provided the camera motion and the intrinsic camera parameters are known. For more details the reader is referred to [Fau93, FLP01].
Note that the camera shift in this experiment has a realistic magnitude. Often the shift is even a bit larger. The human eye distance, for instance, is around 7cm. Nevertheless, already the shift by 2cm leads to rather large displacements of 33 pixels for the foreground object. With a shift by 5cm, the displacements have a magnitude of 80 pixels.

As one can see in Fig. 4.32, the foreground object can easily be distinguished from the background and the disparity is estimated rather well. In other parts of the image, however, the estimates are pretty bad. For instance, the disparity in the foreground region of the table must actually be much larger. Also for parts of the wallpaper, the estimation failed. Like in the last experiment, there are not enough coarse scale structures available in these areas to provide reliable estimates in the presence of such large displacements.

Although the optic flow estimation method yields very accurate estimates in the scope of motion estimation, its accuracy in a typical stereo setting is rather poor. What is wrong?
One reason is that we solved a much harder problem than actually necessary. Assumed the camera parameters are known, the corresponding pixel in the second frame is restricted to be located on a line. This epipolar constraint has been ignored here, though it would reduce the size of the search space considerably. For severe stereo applications, it has to be integrated into the method like proposed in [RD96, ADSW02].
Another reason is the camera arrangement in this experiment. With this arrangement, the focus of the stereo system is on a point somewhere in the mountains, though all objects in the scene are very close. One will obtain much smaller displacements in this scenario, if the camera is rotated such that the foreground object is focused in both frames. This way, disparities in the focused region are close to zero. Also the total magnitude of all displacements in the image is reduced and thus the accuracy increases.

Human motion. An interesting special case of motion in real-world sequences is human motion. Human motion is in general quite complex, since arms and legs often move in different directions than the rest of the body. Fig. 4.33 shows such a scenario where the person tilts to the right while he is raising his right hand (i.e. the hand on the left)[11]. The displacements in this sequence are small in comparison to the desk scenarios, yet most of the other artifacts are also present here.

The result shown in Fig. 4.33 is actually quite good. Despite some blurring, the person's contour is clearly visible and the body as well as the hands move into the right direction.
The blurring can be explained by the coarse-to-fine strategy of the method and the missing small-scale texture in the background: at coarse scales the object boundary is not well localized due to the low resolution of the images, hence the motion of the object is also estimated for pixels belonging partially to the object and partially to the background. Since there is few useful gradient information in the background, the estimated flow is not corrected by successive finer scales. The data term is very small in these areas, thus the smoothness term gets almost full control.

[11] The sequence has been kindly provided by Bodo Rosenhahn, University of Auckland.

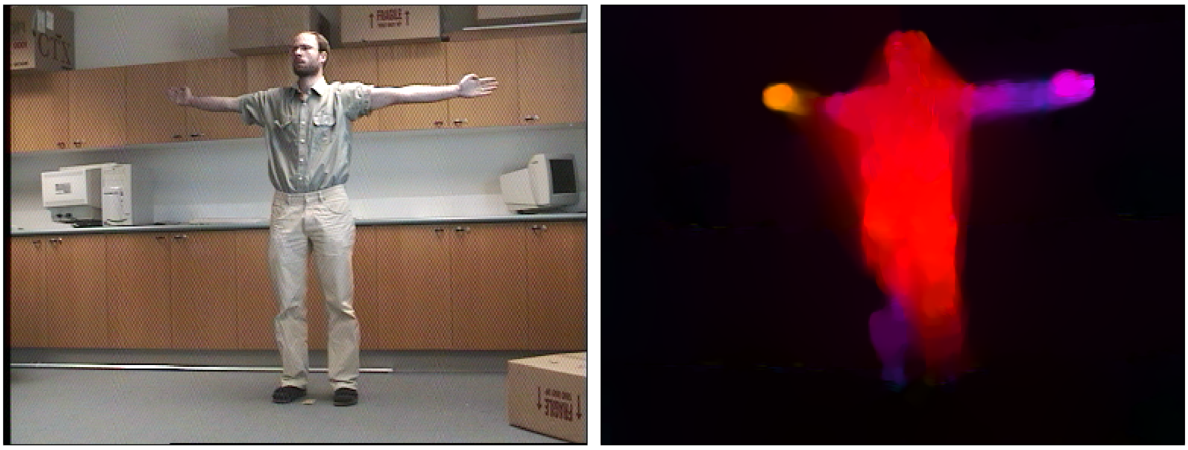


Fig. 4.33: Human motion in a lab environment. The person tilts to the right, raising his left hand. LEFT: First image. RIGHT: Computed displacement field (2-D method, $\sigma = 1$, $\alpha = 50$, $\gamma = 20$).

At large, the experiments have shown that missing image structures arise as the principal remaining problem in optic flow estimation. Even though the model and its numerical implementation are qualified to deal with most kinds of difficulties, the method still depends on a sufficient amount of utilizable information. In this sense optic flow computation is a very typical image analysis task: to improve its performance it is either necessary to increase the amount of information, or to add constraints which restrict the number of possible solutions. This phenomenon will also be faced within the scope of segmentation, which is subject of the next chapter.

5 Image Segmentation

Millions long for immortality, who do not know
what to do with themselves on a rainy Sunday afternoon.

Susan Ertz (1894-1985)
Author

In the preceding chapters, it was often referred to image segmentation as an application of nonlinear diffusion, texture features, and optic flow. Indeed, image segmentation is a topic where many techniques run together. Although the definition of image segmentation as the *partitioning of the image domain into meaningful regions* appears unimpressive at the first glance, the decisive word is 'meaningful'. The signification of 'meaningful' in this definition is still left open, though it is mostly assumed that *a meaningful partitioning extracts the boundaries of the visible objects in the image.* When dealing with this task, image segmentation plays an important role for the interpretation of images. While the techniques in the previous chapters focused on relations between pixels, either in the spatial domain (denoising, texture) or the spatio-temporal domain (optic flow), image segmentation introduces a relation between pixels and objects and therefore establishes a first connection between so far uninterpreted data and an interpreted, abstract, description of the scene.

When considering the extraction of object boundaries, one might think of an edge detector like the gradient magnitude or a more elaborated method like the Canny edge detector [Can86]. However, there are plenty of reasons why an edge detector is not appropriate for extracting object boundaries. Firstly, edges in the image do not always correspond to object boundaries. One might think, for instance, of textured objects. Secondly, edges need not coincide with closed contours. Closed contours, however, are required in order to determine regions and to find a mapping between pixels and objects. Finally, edge detection takes solely local cues into account. In some cases, this can be advantageous, yet in others a global view on the object appearance is more appropriate. It is therefore to be noticed that image segmentation is not the same as edge detection. Image segmentation yields regions, which implies closed contours, not edges.

However, this still leaves the question open how image segmentation can be modelled. What is to be assumed for image segmentation to establish a relation between pixels and objects, i.e., what are the properties of a pixel that make it to belong to one object, while other pixels are assigned to another object? Or to put it in a nutshell: what are the model assumptions?

Usually there are two model assumptions for image segmentation: firstly, there is to be a large similarity of features *within* an object, whereas there should be a large dissimilarity of features *between* objects. Secondly, the object's shape is assumed to be as compact as possible, i.e., the length of the object boundary should be minimal.

It is to be noted that these assumptions about the appearance of objects are not always satisfied. This is why a purely image-driven approach cannot work in all situations, but only if different objects have different appearances or if they move differently. In order to extract objects which basically look similar and move the same way as the background, one has to introduce further assumptions about the objects. The integration of such a knowledge-driven part is briefly discussed versus the end of this chapter. The main focus, however, lies on the image-driven part.

Previous work. The model assumptions stated above are already close to the description by a variational approach. Indeed, a variational formulation with similar model assumptions has been introduced by Mumford and Shah [MS85, MS89]:

$$E(u,\Gamma) = \int_\Omega (u - I)^2 \, \mathbf{dx} + \lambda \int_{\Omega - \Gamma} |\nabla u|^2 \, \mathbf{dx} + \nu \int_\Gamma ds \tag{5.1}$$

where Γ is the set of boundaries that separates the different regions, u is a smooth approximation of the image I within these regions, and λ and ν are positive weighting parameters. A simplified version of the Mumford-Shah functional, which is called the *cartoon limit*, is obtained if $\lambda = 0$. The remaining energy functional

$$E(u,\Gamma) = \int_\Omega (u - I)^2 \, \mathbf{dx} + \nu \int_\Gamma ds \tag{5.2}$$

yields in an optimum a piecewise constant approximation u, and a boundary Γ that separates the constant regions. It is easy to observe that this is in full accordance with the above-mentioned model assumptions for image segmentation: similar neighboring pixels are assigned to the same region, while dissimilar pixels are separated by a region boundary; the total length of region boundaries is sought to be minimal.

While the Mumford-Shah functional already yields a quite general formulation of the image segmentation problem in form of an optimization problem, the way how to minimize the energy has been left open in [MS89]. One of the main problems for minimization is the set of infinitesimal thin boundaries Γ. As a remedy, Ambrosio and Tortorelli [AT92] proposed a regularized version of the functional in (5.1) where the boundaries have finite width. This allows for a gradient descent method to minimize the functional. It can be shown that the approximation of the Mumford-Shah functional resembles nonlinear diffusion with Perona-Malik diffusivity, revealing the relations between nonlinear diffusion and image segmentation. For the cartoon limit in (5.2) another minimization strategy has been suggested in [KLM94, MS94]. This strategy is based on the region merging algorithm introduced in [BGK$^+$89]. Starting with each pixel in the image being a single region, the algorithm successively merges those two regions which lead to the largest energy decrease. The algorithm can thus be regarded as a greedy heuristic for minimizing (5.2). In [MS94] several theoretical properties of this algorithm have been shown, e.g. the elimination of small or thin region and an upper bound for the number of regions. Due to its efficiency and simplicity, the region merging algorithm has become very popular. Although it tends either to over-segmentation or to region boundaries that only match object boundaries if the interior of the object regions is very homogeneous and clearly pops out of the background, for a long period region merging has been one of the best performing segmentation techniques available.

A different approach to image segmentation is represented by so-called *active contour models*, also known as *snakes* [KWT88, Coh91, CC96]. In their original formulation, active contour

models are edge based, i.e., they use an edge detector in order to determine region boundaries. In contrast to a pure edge detector, however, they can ensure closed contours and thus represent regions.
Apart from active contours being edge based, there is a further difference to the Mumford-Shah functional: active contours in their basic formulation are restricted to the extraction of a single object boundary, whereas the Mumford-Shah functional captures arbitrary many object regions. Note that this restriction introduces a severe simplification of the segmentation task, since it implies the knowledge that there is exactly one object to be extracted locally from the image. The optimum number of regions is no longer to be determined.

Special cases of active contours models are *geometric* active contours [CCCD93, MSV95] and *geodesic* active contours [CKS95, KKO$^+$95, KKO$^+$96, CKS97]. Most interesting for the segmentation method derived later in this chapter is the geodesic active contour model, which can be formulated as the minimization problem

$$E(\Gamma) = \int_\Gamma g(|\nabla I|)\, ds. \tag{5.3}$$

with $g(|\nabla I|)$ being an edge detector which yields small values at the position of edges in the image and 1 otherwise. This energy resembles very much the last term of the Mumford-Shah functional. Indeed, minimization of the energy minimizes the boundary length weighted by a value that depends on the existence and steepness of edges. This weight evokes the contour to be attracted by image edges. In contrast to region based segmentation models, such as the Mumford-Shah functional, the interior of the region is not relevant for the result.
In [CBA93] and [Ron94] this strict negligence of region information has been relieved and has finally led to the *geodesic active regions* model [PD99, PD02a] and the Chan-Vese model [CV99, CV01]. Very similar is also the method in [TYW01]. Active contour models, their extensions, and their implementation will be discussed in Section 5.1 in more detail.

If the compactness constraint in image segmentation is ignored, i.e., the length of the boundary is not subject to optimization, then the task comes down to a classical clustering problem as known from statistics. Consequently, many clustering techniques have been applied to image segmentation. The most popular methods are the *normalized cut* [SM97, SM00] and the *mean shift* algorithm [Che95, CM02]. Also *pairwise clustering* [HB97, HPB98] and *semidefinite programming* [KHS04, Keu04] can be applied successfully to image segmentation.
Clustering approaches often include the spatial distance between pixels as an additional dimension in feature space and thereby ensure that the pixel clusters (regions) obey a certain spatial compactness. However, in contrast to most image segmentation methods, pixels assigned to the same cluster need not necessarily be connected in the image. In fact this can be beneficial in cases where the object is occluded by other objects and hence is split into several parts.

While clustering based segmentation approaches deal quite negligently with the spatial relationship of pixels, they are often more sophisticated in the statistical modelling of the regions. While, e.g., the cartoon limit of the Mumford-Shah functional assumes piecewise constant regions, which comes down to modelling the region solely by its mean value, clustering methods in general employ more complex models, such as Gaussian mixture models or nonparametric density estimates. This is beneficial especially in case of textured regions. Modelling the region of a zebra with its black and white stripes by means of the mean value can hardly capture the difference of this region to its background. Modelling the gray value distribution of the region with two Gaussians or the region histogram, on the other hand, leads to a much more precise description of the region's interior.

Segmentation models that include ideas from both the Mumford-Shah functional and clustering methods are based on the *minimum description length (MDL) criterion* [Lec89] and *Bayesian models* [GG84, Win03]. The motivation of MDL comes from information theory. It is to describe the image data D by a model M, such that the number of bits necessary to transmit the data by means of the model is minimal:

$$E(M) = |\mathcal{L}_c(D|M)| + |\mathcal{L}_m(M)| \to \min . \tag{5.4}$$

Thereby $\mathcal{L}_c$ denotes a language for describing the data, given the model, and $\mathcal{L}_m$ is a language for describing the model.
The Bayesian approach has a different motivation. Based on the *maximum a-posteriori probability (MAP) criterion* it seeks to choose the model with the maximum conditional (a-posteriori) probability, given the data. In order to determine this conditional probability, the Bayes rule

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)} \tag{5.5}$$

is applied, where $P(M|D)$ is the a-posteriori probability of the model given the data, $P(D|M)$ is the probability of the data given the model, $P(M)$ is the unconditional probability of the model, and $P(D)$ is the unconditional probability of the data. Maximizing $P(M|D)$ with respect to the model leads to the energy minimization problem:

$$E(M) = -P(D|M)P(M) \to \min \tag{5.6}$$

or equivalently to

$$E(M) = -\log P(D|M) - \log P(M) \to \min . \tag{5.7}$$

The MDL criterion and the MAP criterion look very similar, and indeed there are many relations [Lec89, ZY96]. Furthermore, both criterions are so abstract that they can capture many other segmentation models. When choosing appropriate languages $\mathcal{L}_c$ and $\mathcal{L}_m$ in (5.4), for instance, one can obtain the Mumford-Shah functional. The same can be achieved with (5.6) if the set of possible models is restricted. Thus it is not surprising that many implementations under the logo of MDL or Bayes come down to region merging algorithms or active contours.

The *region competition* approach presented in [ZY96] has exploited this fact and has integrated ideas from most of the previously mentioned techniques. It is sought to minimize an energy that is more general than the Mumford-Shah functional or the active contour model, but less abstract than the MDL or MAP criterion. The optimization problem is stated by the energy functional

$$E(\Omega_i, p_i, N) = \sum_{i=1}^{N} \left(-\int_{\Omega_i} \log p_i \, \mathbf{dx} \; + \; \frac{\nu}{2} \int_{\Gamma_i} ds \; + \; \varrho \right) \tag{5.8}$$

with side conditions $\bigcup_i \Omega_i = \Omega$ and $\bigcap_i \Omega_i = \emptyset$. The unknowns contain the number of regions N, the regions Ω_i with their boundaries Γ_i, and the probability density functions $p_i \equiv P(D|M_i)$ within the regions. The energy consists of three terms weighted relative to each other by the parameters $\nu \geq 0$ and $\varrho \geq 0$. The first term captures the first factor of the MAP criterion in (5.6). It is equivalent to the first term of the cartoon limit of the Mumford-Shah functional, if the distribution model respects merely the region's mean value. The second term introduces a penalty on the length of the region boundaries, like in the Mumford-Shah functional. It is also present in many active contours models. The third term finally supplements a penalty for the number of regions.

For minimizing this rather complex energy, an algorithm is proposed in [ZY96], which is based on region growing and region merging ideas. It further contains concepts from active contour models. Region growing has been suggested in [AB94]. It is based on first placing manually some so-called 'seeds', i.e. pixels that define the origin of regions, into the image. Pixels at a region boundary join the region, if their values fit well to the region model and if they have not yet been assigned to any other region. Iterating this procedure results in successively growing regions. This algorithm is fully adopted in [ZY96]. Additionally, colliding regions are allowed to compete for pixels. Thus the boundaries between regions can move in order to minimize the energy, like in the case of active contours. The algorithm in [ZY96] further allows for merging two regions, if this decreases the total energy. The latter is a concept stemming from the region merging algorithm.

The list of existing segmentation methods presented here is not complete, but mentions only some of the most popular techniques which are related to the technique proposed in this chapter. Simple algorithmically motivated techniques like thresholding after some preprocessing of the image have been left out here. Also the morphological method of watersheds [BL79, Jac96] is not discussed. Another segmentation method based on algebraic multi-grid should only be mentioned briefly, though it provides very good segmentation results [SBB00, SBB01, GSBB03].

Contributions. The segmentation method described in this chapter seeks to minimize a similar energy as in (5.8). In contrast to the work in [ZY96], however, the minimization algorithm is based on implicit active contour models which represent regions and their boundaries by means of the *level set framework* [DT79, OS88]. This framework, which is discussed in detail in the first section of this chapter, conveniently allows to deal with the second term of the functional. Different statistical models for describing the region's interior are discussed. Besides the Gaussian distribution used in [ZY96], this includes a nonparametric Parzen density estimate [Ros56, Par62] and a new model with local statistics. Moreover, since optimization problems in image segmentation often contain many local optima, a coarse-to-fine minimization strategy is introduced for increasing the number of situations where the method runs into the global optimum.

In Section 5.2, texture and motion features from the previous chapter are integrated into the segmentation model. Section 5.3 then lifts the restriction to split the image into exactly two regions and therefore allows the extraction of multiple object contours in an image.

The segmentation techniques described in Section 5.1 - 5.3 are purely image-driven. Despite all efforts to respect as much image information as possible by including texture and motion cues, this information is often not sufficient to extract satisfactory object contours. In the last section of this chapter, the model is hence extended by a knowledge-driven part, following the basic concepts in [LGF00, RP02, Cre02, COS04]. In contrast to these earlier works, which rely on two-dimensional shape knowledge, it is suggested here to employ three-dimensional shape knowledge, which has the advantage of modelling perspective views from the objects in the real, three-dimensional world.

Most of the work in Section 5.1 and Section 5.2 has been in collaboration with Mikaël Rousson. Section 5.4 is based on joint work with Bodo Rosenhahn.

5.1 Level Sets for Segmentation

The segmentation model of implicit active contours is based on level set methods, which actually originate from computational fluid dynamics [DT79, OS88]. Meanwhile, the level set framework has become a very popular tool for curve representation in image analysis.

Let $C(s)$ be a closed curve parameterized with the arc-length s. Such a curve can be represented by a so-called *embedding function* or *level set function* $\Phi(x, y)$ when introducing C as a level line into Φ [OS88]. This can be done, e.g., by a signed distance function

$$\Phi(\mathbf{x}) := \begin{cases} -\text{dist}(\mathbf{x}, C) & \text{if } \mathbf{x} \text{ outside the area encircled by } C \\ \ \ \text{dist}(\mathbf{x}, C) & \text{else} \end{cases} \tag{5.9}$$

where $\text{dist}(\mathbf{x}, C)$ denotes the distance of a point x to the next point on the curve C, provided a certain distance measure (e.g. Euclidean distance). The zero-level line of Φ is then equivalent to the curve C.

With this kind of curve representation it is possible to describe a curve evolution of type

$$\partial_t C = \beta \mathbf{n} \tag{5.10}$$

by a partial differential equation that acts on Φ:

$$\partial_t \Phi = \beta \cdot |\nabla \Phi| \tag{5.11}$$

whereas $\mathbf{n}(s)$ describes the outer normal direction of the curve C at s and $\beta(s)$ is a scalar force that evolves the curve in direction of $\pm\mathbf{n}$. In a similar way, a surface $S(r(x, y, z), s(x, y, z))$ can be embedded into a three-dimensional level set function $\Phi(x, y, z)$.

What is the advantage of such an embedding? On the first glance, it only turns a one-dimensional structure $C(s)$ into a two-dimensional structure $\Phi(x, y)$ and therefore increases the computational effort.

- One advantage becomes apparent when considering spatial discretizations of C and Φ. While Φ always stays on a fixed homogeneous grid, independent from the evolution of its level lines, the grid points in a discretization of C move together with the evolving curve. This can evoke severe differences in the distance between grid points and requires a regridding during the evolution. Since (5.11) keeps a fixed coordinate system, it is also called the *Eulerian formulation* of (5.10) [Set90].

- The second and more important advantage is the representation not only of the curve itself, but also of the region encircled by this curve. By the sign of $\Phi(\mathbf{x})$ one can immediately see whether $\mathbf{x}$ is inside or outside the curve. This is especially important in image segmentation where often both the region boundaries Γ_i and the encircled regions Ω_i play a role in the optimization process.

- As a third advantage, the level set representation allows for topological changes. Consider a region that is split into two parts that are not connected. The representation of such a region by a curve C is not immediate. The most convenient solution might be to use two curves C_1 and C_2 to represent the region. But what will happen, if the evolution drives the two parts towards each other, such that they finally merge? By embedding the region boundaries as a level line into the level set function Φ, it is not necessary to care about such topological changes as they are captured implicitly by this kind of representation. In the scope of image segmentation this offers the advantage that regions need not necessarily be connected. This is in contrast to most other image segmentation algorithms and provides a property mainly known from clustering approaches.

5.1.1 Mean Curvature Motion and Implicit Active Contours

The level set framework is the basis of implicit active contours as introduced in [CCCD93, MSV95, CKS95, KKO+95]. The implicit representation of the contour via a level set function thereby allows for a convenient minimization of the contour length while the curve is simultaneously aligned to edges in the image.

Mean curvature motion. Minimizing the curve length can be performed by so-called *mean curvature motion (MCM)* [ES91, AGLM93, ST93]. It can be described explicitly by the *geometric heat equation* or *Euclidian shortening flow* [Gag86, GH86, Gra87, ES91]

$$\partial_t C = \kappa \mathbf{n} \tag{5.12}$$

where $\kappa = \frac{\det(C_s, C_{ss})}{|C_s|^3}$ is the curvature of C. The equivalent implicit formulation in the level set framework reads:

$$\partial_t \Phi = |\nabla \Phi| \operatorname{div}\left(\frac{\nabla \Phi}{|\nabla \Phi|}\right). \tag{5.13}$$

One can show that mean curvature motion minimizes the Euclidian arc-length of the curve C, i.e., it minimizes the energy

$$E(C) = \oint |C_s(s)|\, ds = \int_C ds \tag{5.14}$$

Since usually one of the model assumptions for image segmentation is a minimum boundary length between regions, this energy term is part of many variational approaches for image segmentation, among others the Mumford-Shah functional and the functional of Zhu-Yuille in (5.8). The implicit formulation of the boundary constraint by means of the level set framework now provides also a sound way to perform the minimization of such energy functionals, namely by means of mean curvature motion.

In order to implement mean curvature motion, it is useful to write (5.13) as:

$$\partial_t \Phi = \frac{\Phi_y^2 \Phi_{xx} - 2\Phi_x \Phi_y \Phi_{xy} + \Phi_x^2 \Phi_{yy}}{\Phi_x^2 + \Phi_y^2}. \tag{5.15}$$

Spatial discretization of the partial derivatives at pixels (i, j)

$$(\Phi_{i,j})_x \approx \frac{\Phi_{i+1,j} - \Phi_{i-1,j}}{2}$$

$$(\Phi_{i,j})_y \approx \frac{\Phi_{i,j+1} - \Phi_{i,j-1}}{2}$$

$$(\Phi_{i,j})_{xx} \approx \Phi_{i+1,j} - 2\Phi_{i,j} + \Phi_{i-1,j}$$

$$(\Phi_{i,j})_{yy} \approx \Phi_{i,j+1} - 2\Phi_{i,j} + \Phi_{i,j-1}$$

$$(\Phi_{i,j})_{xy} \approx \begin{cases} \frac{2\Phi_{i,j}+\Phi_{i-1,j-1}+\Phi_{i+1,j+1}-\Phi_{i-1,j}-\Phi_{i+1,j}-\Phi_{i,j-1}-\Phi_{i,j+1}}{2} & \text{if } (\Phi_{i,j})_x(\Phi_{i,j})_y < 0 \\ \frac{\Phi_{i-1,j}+\Phi_{i+1,j}+\Phi_{i,j-1}+\Phi_{i,j+1}-2\Phi_{i,j}-\Phi_{i-1,j+1}+\Phi_{i+1,j-1}}{2} & \text{else} \end{cases}$$

and time discretization with forward differences leads to the update equation

$$\Phi_{i,j}^{k+1} = \begin{cases} \Phi_{i,j}^k + \tau \frac{(\Phi_{i,j}^k)_y^2 (\Phi_{i,j}^k)_{xx} - 2(\Phi_{i,j}^k)_x (\Phi_{i,j}^k)_y (\Phi_{i,j}^k)_{xy} + (\Phi_{i,j}^k)_x^2 (\Phi_{i,j}^k)_{yy}}{(\Phi_{i,j}^k)_x^2 + (\Phi_{i,j}^k)_y^2} & \text{if } |\nabla \Phi_{i,j}^k| = 0 \\ \Phi_{i,j}^k & \text{else} \end{cases} \tag{5.16}$$

with iteration index k and time step size τ. There is no stability theory for this scheme, yet experimental evaluation reveals stable results for $\tau \leq 0.25$.

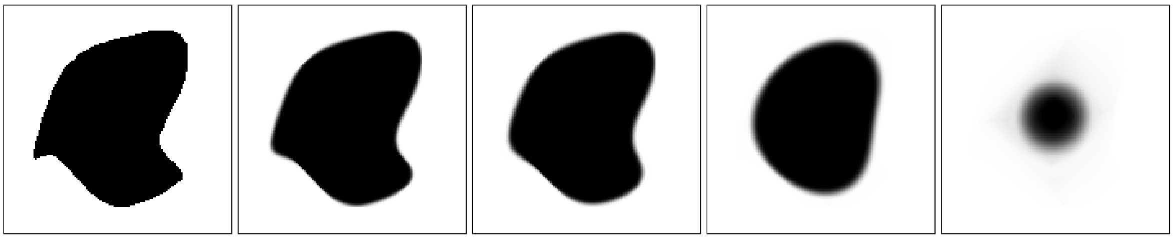


Fig. 5.1: Illustration of mean curvature motion.
FROM LEFT TO RIGHT: Initial Φ and Φ after $t = 20$, $t = 50$, $t = 250$, and $t = 1000$.

Fig. 5.1 illustrates the behavior of mean curvature motion. One can see that the implicit curve is propagated in inner normal direction with a speed dependent on its curvature. Parts with a high curvature evolve faster than parts with a small curvature. The curve shrinks within finite time to a circular point.

Geodesic active contours. Based on the curve length minimization with mean curvature motion, geodesic active contours additionally introduce an edge detector $g(|\nabla I|)$, which performs the task to draw the curve towards edges in the image I [CKS95, KKO$^+$95]. Like the diffusivity functions in Chapter 2, $g(s) > 0$ is a decreasing function, i.e., it becomes small in the presence of edges in the image. A possible choice is, for instance

$$g(|\nabla I|) = \frac{1}{\sqrt{|\nabla I|^2 + \epsilon^2}} \tag{5.17}$$

with $\epsilon = 0.1$. This edge indicator can be added to the evolution equation in form of a weighting factor:

$$\partial_t \Phi = |\nabla \Phi| \operatorname{div} \left(g(|\nabla I|) \frac{\nabla \Phi}{|\nabla \Phi|} \right). \tag{5.18}$$

One can show [CKS97] that this equation is the steepest descent for minimizing the energy

$$E(C) = \oint g(|\nabla I|) \, |C_s(s)| \, ds = \int_C g(|\nabla I|) \, ds. \tag{5.19}$$

and empirically converges to a non-trivial steady state. In (5.19) it is easy to see that in a steady state the curve length is locally minimized with respect to some image-induced metric defined by $g(|\nabla I|)$.

Applying the chain rule, (5.18) can be rewritten as

$$\partial_t \Phi = g \, |\nabla \Phi| \operatorname{div} \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) + \nabla g^\top \nabla \Phi. \tag{5.20}$$

It becomes obvious that the evolution equation actually consists of two terms. The first one performs mean curvature motion weighted by the edge indicator function g. With solely this term, the curve would still shrink to a circular point for $t \to \infty$, since $g > 0$. The shrinkage would only be slowed down in the presence of edges, yet it could not be stopped. Only the second term in the evolution equation can prevent further shrinkage, as it evolves the curve in direction of smaller g. With this term, the curve can even evolve in outer normal direction, provided the attracting image edge is sufficiently steep and close enough to the evolving contour. In active contour models besides geodesic active contours, this property has to be introduced artificially by adding a balloon force in outer normal direction, which involves a weighting parameter [Coh91].

This already indicates the dependence of the outcome from the initialization of the contour. The evolution equation (5.18) is only a *local* optimization method for minimizing (5.19). Consequently, it can yield only the next local optimum. This means that the curve is attracted by the next relevant edge in the image, although an edge further away from the initial curve could yield a smaller weighted curve length.

The geodesic active contour model can be implemented by means of (5.20). Its first term can be discretized like mean curvature motion with an additional weighting factor. For a stable implementation of the second term, a so-called *upwind scheme* using one-sided differences can be applied [OS88]. Such schemes originate again from computational fluid dynamics[1]. The update equation concerning the second term in (5.20) implemented by means of the upwind scheme reads:

$$\begin{aligned}\Phi^{k+1}_{i,j} = \Phi^k_{i,j} \;&+\; \tau\left(\max\left(0, \tfrac{g_{i+1,j}-g_{i-1,j}}{2}\right)(\Phi^k_{i+1,j}-\Phi^k_{i,j}) + \min\left(0, \tfrac{g_{i+1,j}-g_{i-1,j}}{2}\right)(\Phi^k_{i,j}-\Phi^k_{i-1,j})\right)\\ &+\; \tau\left(\max\left(0, \tfrac{g_{i,j+1}-g_{i,j-1}}{2}\right)(\Phi^k_{i,j+1}-\Phi^k_{i,j}) + \min\left(0, \tfrac{g_{i,j+1}-g_{i,j-1}}{2}\right)(\Phi^k_{i,j}-\Phi^k_{i,j-1})\right)\end{aligned} \tag{5.21}$$

with iteration index k and time step size τ. Experimental stability is determined by the first term. The whole scheme has turned out to yield stable results for $\tau \leq \frac{\epsilon}{4}$.

Unfortunately, geodesic active contours implemented by this explicit scheme converge comparatively slowly. In order to speed up the curve evolution, several ideas have been suggested. One of these ideas is the *narrow band* approach [Cho93, AS95, PMO$^+$99, PD00b]. It is based on the fact that the update in the basic implementation is performed on the whole image domain, although only a small part of this domain, which is close to the propagating curve, is of interest. The narrow band approach thus reduces computations to a small area around the zero-level line of Φ, the narrow band.

An alternative strategy to speed up the process is to replace the explicit time discretization by a semi-implicit scheme which is stable for arbitrary large time steps. Such semi-implicit schemes, like the AOS scheme, can decrease the computational costs by a factor 10 or more, see Section 2.1.3. The AOS scheme has been proposed for implicit active contour models in [GKRR01] and [Wei01], whereas the advantage of the method in [Wei01] lies in the fact that it needs no reinitialization of the level set function to the signed distance function after each time step.

Finally, also multi-scale strategies are a possible way to speed up active contours [PD00b]. One starts the contour evolution on a downsampled version of the image and uses the resulting curve as initialization for the evolution with the original image size. Apart from faster computation, this strategy is in the sense of continuation methods and can additionally help to avoid being trapped by local minima. This issue is discussed later on in more detail.

Fig. 5.2 shows the performance of the geodesic active contour model. Initialized with a rectangle close to the object to be extracted, the contour moves towards the closest relevant edges. Note that due to the second term in (5.20) the contour converges to a steady state. Moreover, the contour can capture non-convex parts of the object and evolve in outer normal direction. Obviously, the energy for aligning the curve at the edges of the sheep's legs is larger than the energy for the shortcut. Thus the desired segmentation cannot be fully provided by the model. One can also see that the edge along the neck of the mother sheep attracts the curve, although the edge along the neck of the lamb appears to be steeper. This is a typical case of a local minimum. With an initialization closer to the object contour one obtains a result, which is closer to the desired segmentation, see Fig. 5.3.

[1] See Chapter 3.

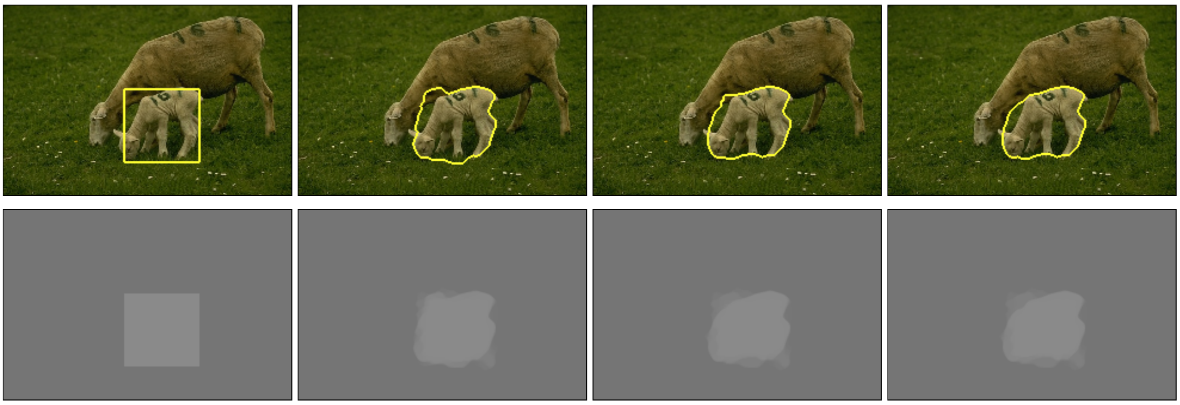


Fig. 5.2: Segmentation with the geodesic active contour model.
Top Row from Left to Right: Curve after $t = 0$, $t = 1000$, $t = 5000$, and $t = 10000$ (steady state).
Bottom Row: Level set function Φ.

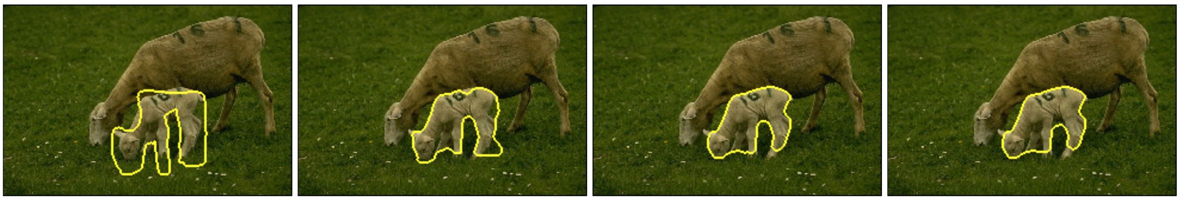


Fig. 5.3: Segmentation with the geodesic active contour model and a different initialization.
From Left to Right: Curve after $t = 0$, $t = 1000$, $t = 3000$, and $t = 10000$ (steady state).

In conclusion, the geodesic active contour model has several advantages:

- Its main advantage is the sound minimization of the energy functional by means of the discretized steepest descent equation. There are no algorithmic supplements like regridding necessary.
- Additionally, the implicit curve representation allows for topological changes. Hence, partially occluded objects that are split into several parts can be handled without additional efforts.
- The energy functional is very simple and contains only a few parameters, this is the choice of the edge detector and the edge weighting function g.

Apart from this, however, the model has also some shortcomings:

- First of all, the interior of the extracted regions is completely neglected. The only cues that drive the evolution are the edges in the image. However, edges are not very reliable for extracting objects. Especially in the presence of textured objects, edges do hardly contain any useful information for the segmentation.
- Since the evolution is drawn to the next significant edge in the image, any edge in between the initialization and the sought object contour can attract the solution. While in supervised medical segmentation applications this strong dependence on the initialization might be considered as beneficial, it is certainly a severe disadvantage in unsupervised segmentation.

The goal of the technique described in the next section is to keep the advantages of geodesic active contours while addressing its shortcomings.

5.1.2 Region Based Active Contours

Both problems that appear with the active contour model originate from the negligence of region information. While the edge detector in the model ensures that the data on both sides of the contour is as dissimilar as possible, it cannot deal with the requirement that also the interior of a region should be as homogeneous as possible.

Region based propagation forces. An immediate consequence of these considerations is to include region information in the active contour model, as proposed in [CBA93, Ron94]. This concept has become very popular in connection with the *geodesic active region* model of Paragios and Deriche [PD99, PD02b, PD02a] and the model of Chan and Vese [CV99, CV01]. The latter model even neglects the edge information completely and presents itself as a level set based formulation of the Mumford-Shah functional. If the image domain Ω is split into two regions Ω_1 and Ω_2, with $\Omega_1 \cup \Omega_2 = \Omega$ and $\Omega_1 \cap \Omega_2 = \emptyset$, the cartoon limit of the Mumford-Shah functional can be written as

$$E(\Gamma) = \int_{\Omega_1} (I - \mu_1)^2 \, \mathbf{dx} + \int_{\Omega_2} (I - \mu_2)^2 \, \mathbf{dx} + \nu \int_{\Gamma} ds \tag{5.22}$$

where μ_1 and μ_2 denote the mean values of the two regions and Γ stands for the boundary between them. The weighting parameter $\nu \geq 0$ is the same as in the Mumford-Shah functional. Making use of a level set function $\Phi(\mathbf{x})$, with $\Phi(\mathbf{x}) \geq 0$ if $\mathbf{x} \in \Omega_1$ and $\Phi(\mathbf{x}) < 0$ if $\mathbf{x} \in \Omega_2$, the zero-level line of Φ marks the region boundary Γ and (5.22) can be expressed as:

$$E(\Phi) = \int_{\Omega} \Big(H(\Phi)(I - \mu_1)^2 + (1 - H(\Phi))(I - \mu_2)^2 + \nu \, |\nabla H(\Phi)| \Big) \, \mathbf{dx} \tag{5.23}$$

where $H(s)$ denotes a regularized version of the Heaviside function, i.e., $\lim_{s \to -\infty} = 0$, $\lim_{s \to \infty} = 1$, and $H(0) = 0.5$. Often a function based on the arc tangent function or the error function is used for H.

The formulation in (5.23) allows for an energy minimization by means of the gradient descent

$$\partial_t \Phi = H'(\Phi) \left((I - \mu_2)^2 - (I - \mu_1)^2 + \nu \operatorname{div} \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) \right) \tag{5.24}$$

and updates on the means μ_1 and μ_2 in Ω_1 and Ω_2, respectively:

$$\mu_1 = \frac{\int_{\Omega} I \, H(\Phi) \, \mathbf{dx}}{\int_{\Omega} H(\Phi) \, \mathbf{dx}} \qquad \mu_2 = \frac{\int_{\Omega} I \, (1 - H(\Phi)) \, \mathbf{dx}}{\int_{\Omega} (1 - H(\Phi)) \, \mathbf{dx}}. \tag{5.25}$$

The first two terms in (5.24) are easy to implement. The last term looks like TV flow. In combination with $H'(\Phi)$, however, it is better implemented by means of mean curvature motion restricted to a narrow band, as with a time rescaling $H'(\Phi)$ can be replaced by $|\nabla \Phi|$ [ZCMO96]. This way, the nature of (5.24) as curve evolution in normal direction becomes apparent:

$$\partial_t C = \beta \mathbf{n} \qquad \text{with} \qquad \beta = (I - \mu_2)^2 - (I - \mu_1)^2 + \nu \operatorname{div} \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right). \tag{5.26}$$

The evolution equation propagates the curve in a way that it separates two regions with different mean values. Being a special case of the cartoon model of the Mumford-Shah functional thereby explains immediately why there is no edge term necessary for the method to work.

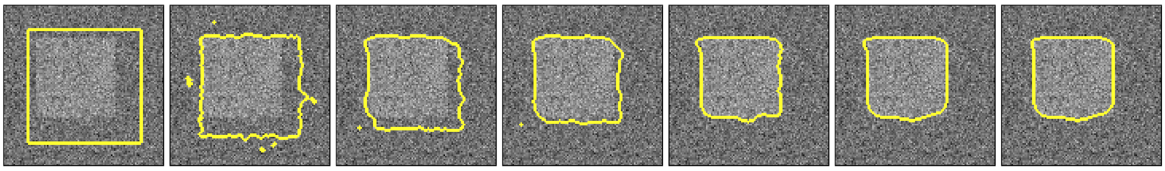


Fig. 5.4: Segmentation with the Chan-Vese model ($\nu = 40$).
FROM LEFT TO RIGHT: Curve after $t = 0$, $t = 1$, $t = 2$, $t = 3$, $t = 4$, $t = 10$, and $t = 20$ (steady state).

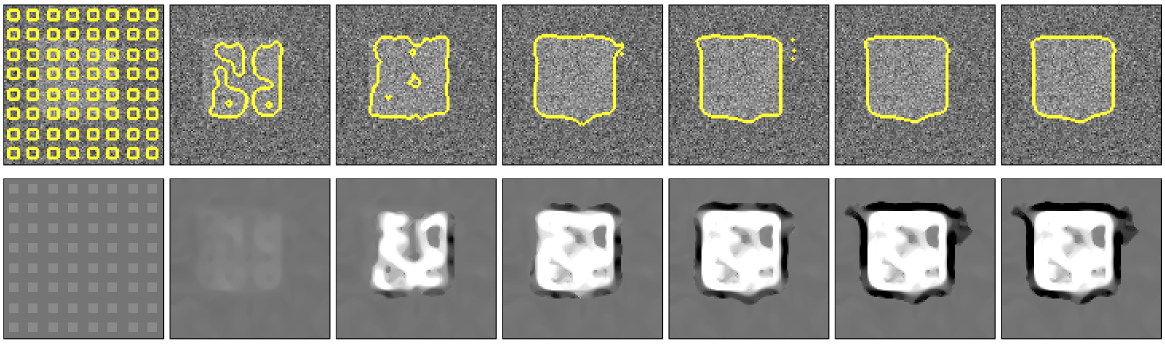


Fig. 5.5: Segmentation with the Chan-Vese model and a different initialization ($\nu = 40$).
FROM LEFT TO RIGHT: Curve after $t = 0$, $t = 1$, $t = 2$, $t = 3$, $t = 4$, $t = 10$, and $t = 20$ (steady state).
BOTTOM ROW: Level set function Φ.

Fig. 5.4 and Fig. 5.5 demonstrate the curve evolution with the Chan-Vese model. Since the model is less dependent on image edges but is driven by the region information, it yields satisfactory results also with initializations that are far away from the optimum solution.
One can also see that the curve evolves much faster than in edge based active contour models. In this example, a steady state is already reached after an evolution time of 20, while evolution times for edge-based active contours are at least two orders of magnitude larger. Since the time step sizes are equal and each iteration induces similar computational costs, the region based active contour model is usually much faster.

In the bottom row of Fig. 5.5 one can see that the region based forces lead to unbounded values in the level set function as soon as it is clear to which region a pixel belongs. This eventually evokes very steep slopes of the level set function along the zero-level line. Moreover, the large values may inhibit the zero-level line to move. To avoid these effects, it has been proposed to constrain the slope of the level set function to $|\nabla\Phi| = 1$ [GF00]. However, it is already sufficient to restrict the range of the values of Φ by clipping values that are smaller or larger than certain thresholds in order to obtain an efficient evolution of the curve.

It may further arise the question about the importance of the length constraint, i.e., the choice of the smoothness parameter ν. In fact, the length constraint is very important to avoid the isolation of noisy pixels, as demonstrated in Fig. 5.6. If ν is chosen too large, corners of objects get rounded, yet one can see that a quite large range of values of ν leads to satisfactory results.

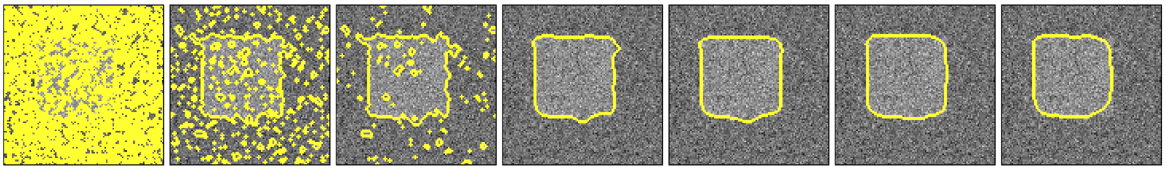


Fig. 5.6: Segmentation with the Chan-Vese model and different settings for ν.
FROM LEFT TO RIGHT: $\nu = 0$, $\nu = 2$, $\nu = 4$, $\nu = 20$, $\nu = 40$, $\nu = 200$, and $\nu = 800$.

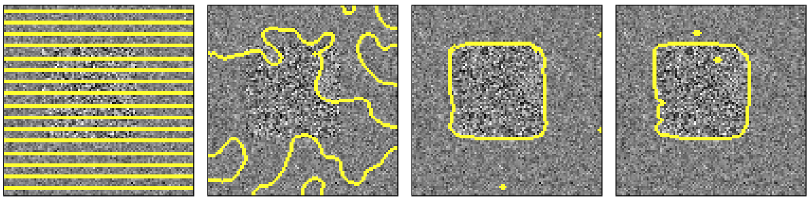


Fig. 5.7: Synthetic image, where regions have same mean but different standard deviations. FROM LEFT TO RIGHT: Initialization. Result with the piecewise constant model ($\nu = 40$), the Gaussian model ($\nu = 0.4$), and the nonparametric model ($\nu = 0.4$, $\sigma_h = 8$).

Region statistics. An important question that arises with region based active contours is how to model the interior of regions. The Chan-Vese model based on the Mumford-Shah functional employs actually the simplest possible model, namely the model of piecewise constant regions. One can certainly think of cases where this model is not appropriate to distinguish the particular regions. An example is shown in Fig. 5.7 where the two regions contain Gaussian noise with the same mean but different standard deviation. In such a case, where the mean value is not sufficient to distinguish the regions, segmentation with the piecewise constant model must fail.

A reasonable way to find a remedy for this shortcoming is to extend the complexity of the region statistics. According to the maximum a-posteriori criterion mentioned in the introduction of this chapter, the segmentation has to maximize the a-posteriori probability $P(M|D)$ of the model M given the data D. Reinterpreting this criterion in the notation of the active contour model, the model M is one of the regions, i.e. either Ω_1 or Ω_2, and the data D is the image gray value I. The Bayes rule allows to express $P(\mathbf{x} \in \Omega_i | I(\mathbf{x}))$, $i = \{1, 2\}$, as

$$P(\mathbf{x} \in \Omega_i | I(\mathbf{x}) = s) = \frac{P(I(\mathbf{x}) = s | \mathbf{x} \in \Omega_i) P(\mathbf{x} \in \Omega_i)}{P(I(\mathbf{x}) = s)}. \tag{5.27}$$

While $P(\mathbf{x} \in \Omega_i)$ is the a-priori probability of region Ω_i, which allows to integrate further assumptions about the region besides its region statistics, $p_i(s) := P(I(\mathbf{x}) = s | \mathbf{x} \in \Omega_i)$ is the probability density in region Ω_i, i.e., the gray value distribution within this region. The a-priori probability of the image gray values $P(I(\mathbf{x}) = s)$ is independent of the choice of the region and can therefore be neglected for optimization.

The only a-priori knowledge about the regions that is available at this point is the length constraint on the object contour. Supplementing the assumption that the gray values of particular image pixels are independent, which is reasonable as long as we do not know the kind of interdependence between the pixels, this yields the task to maximize

$$\prod_{\mathbf{x} \in \Omega_1} p_1(\mathbf{x}) \prod_{\mathbf{x} \in \Omega_2} p_2(\mathbf{x}) \;\; e^{-\nu \int_\Gamma ds} \tag{5.28}$$

or equivalently to minimize

$$E(\Gamma) = -\int_{\Omega_1} \log p_1(\mathbf{x}) \, \mathbf{dx} - \int_{\Omega_2} \log p_2(\mathbf{x}) \, \mathbf{dx} + \nu \int_\Gamma ds. \tag{5.29}$$

This energy functional is very similar to those proposed in [ZY96] and [PD02a]. It can further be expressed by means of the level set framework [RD03, RBD03]:

$$E(\Phi) = \int_\Omega \Big(- H(\Phi) \log p_1 - (1 - H(\Phi)) \log p_2 + \nu |\nabla H(\Phi)| \Big) \, \mathbf{dx}. \tag{5.30}$$

In fact, the Chan-Vese model in (5.23) is only a special case of this energy functional. A typical choice to model the probability density functions is a Gaussian function with mean μ and standard deviation σ:

$$p(s) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(s-\mu)^2}{2\sigma^2}}. \tag{5.31}$$

Modelling the region interior with such a probability density function and a fixed standard deviation of $\sigma = \sqrt{0.5}$, leads exactly to (5.23).

Since the energy in (5.30) is formulated by means of a level set function, minimization can be performed by a gradient descent according to the Euler-Lagrange equations of (5.30):

$$\partial_t \Phi = H'(\Phi) \left(\log p_1 - \log p_2 + \nu \operatorname{div} \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) \right). \tag{5.32}$$

Like the mean values in (5.24), also the probability densities p_1 and p_2 have to be updated after each iteration. For the Gaussian model in (5.31) this comes down to updating

$$\begin{aligned} \mu_1 &= \frac{\int_\Omega I\, H(\Phi)\, \mathbf{dx}}{\int_\Omega H(\Phi)\, \mathbf{dx}} & \mu_2 &= \frac{\int_\Omega I\, (1 - H(\Phi))\, \mathbf{dx}}{\int_\Omega (1 - H(\Phi))\, \mathbf{dx}} \\ \sigma_1 &= \sqrt{\frac{\int_\Omega (I - \mu_1)^2\, H(\Phi)\, \mathbf{dx}}{\int_\Omega H(\Phi)\, \mathbf{dx}}} & \sigma_2 &= \sqrt{\frac{\int_\Omega (I - \mu_2)^2\, (1 - H(\Phi))\, \mathbf{dx}}{\int_\Omega (1 - H(\Phi))\, \mathbf{dx}}}. \end{aligned} \tag{5.33}$$

As demonstrated in Fig. 5.7, there are cases where such a more complex statistical model is beneficial. In contrast to the piecewise constant model, the Gaussian probability density function in (5.31) is capable to capture the boundary between the two regions.

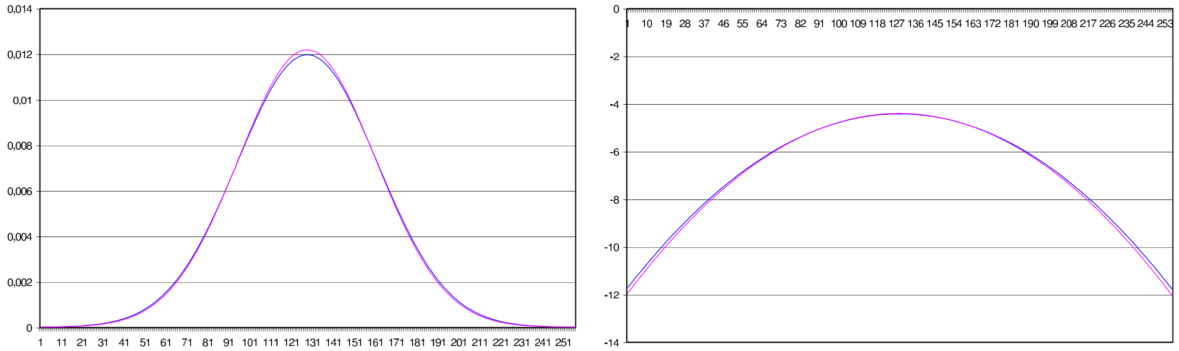


Fig. 5.8: Gaussian probability densities for the image in Fig. 5.7 estimated for the initialization. LEFT: BLUE LINE: p_1. MAGENTA LINE: p_2. RIGHT: BLUE LINE: $\log p_1$. MAGENTA LINE: $\log p_2$.

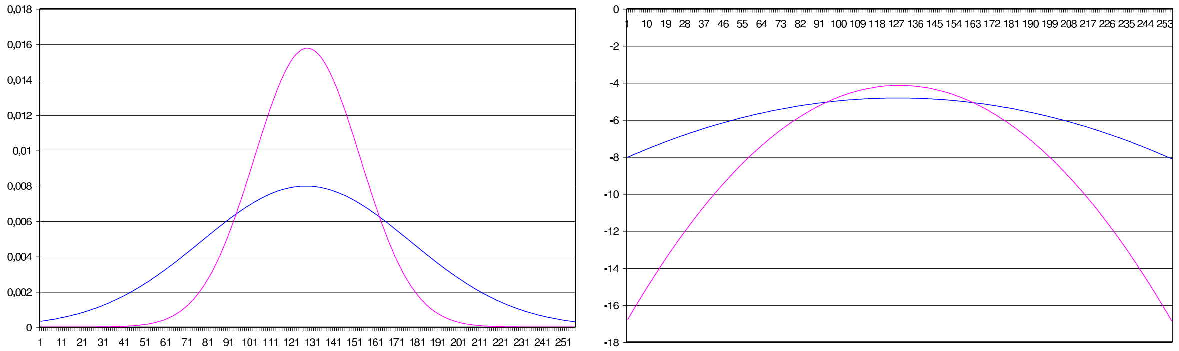


Fig. 5.9: Gaussian probability densities for the image in Fig. 5.7 in the final state. LEFT: BLUE LINE: p_1. MAGENTA LINE: p_2. RIGHT: BLUE LINE: $\log p_1$. MAGENTA LINE: $\log p_2$.

Fig. 5.8 and Fig. 5.9 reveal the way how the method captures the region statistics by the evolution process. Although the curve initialization is a very bad guess, it yields slightly different probability densities in the two regions. Thus for each specific gray value in the image, the assignment of the pixel to one region is more probable than to the other. This difference in the probability drives the evolution of the curve in (5.32). The curve evolution, on the other hand, causes the difference in the probability density functions to grow until the optimum assignment of all pixels is reached. As a result, one region captures the area with small standard deviation, while the other region takes care of the rest.

Nonparametric density estimates. Modelling the gray value distributions of the regions by a Gaussian function is only one of many possibilities [JK69, Sil86]. The full Gaussian model is certainly more general than a model solely based on the mean value. However, one may wonder if the model is general enough. Even if the Gaussian function is usually able to distinguish two different distributions, it may not be capable to properly represent the distribution of values measured in the regions.
Consider, for instance the stripes of a zebra. They mainly contain black and white pixels. The Gaussian function, however, cannot capture this distribution. It will estimate some shade of gray as mean and a large standard deviation.
Especially when the segmentation model is extended to more than one feature channel, as this is the case in Section 5.2, it is important that the probability densities of channels that are not useful for discrimination at least coincide with the data in the regions. Otherwise those channels could disturb the discrimination by means of other, more useful, feature channels.

Thus it is proposed here to employ a nonparametric density estimate [Ros56], in particular a Parzen density estimate [Par62], which can describe the region statistics much more accurately than a Gaussian estimate. A similar procedure has been suggested in [KFY$^+$02]. Instead of estimating the probability density by computing the parameters of a parametric function, the Parzen density estimate employs the region histogram smoothed by a kernel function, which is chosen here to be the Gaussian kernel K_{σ_h} with standard deviation σ_h:

$$p_i(s) = K_{\sigma_h} * \frac{\int_{\Omega_i} \mathbf{1}_{\{I=s\}}\,\mathbf{dx}}{\int_{\Omega_i} \mathbf{dx}} \qquad \mathbf{1}_{\{I=s\}} := \begin{cases} 1 & \text{if } I(\mathbf{x}) = s \\ 0 & \text{else} \end{cases} \tag{5.34}$$

Fig. 5.7 shows that also the Parzen density estimate can discriminate the two regions in this test image. Fig. 5.10 depicts the corresponding probability densities of the two regions in the steady state. In this case they resemble very much the Gaussian estimates, which is not surprising as the noise in the two regions is indeed a Gaussian distribution.

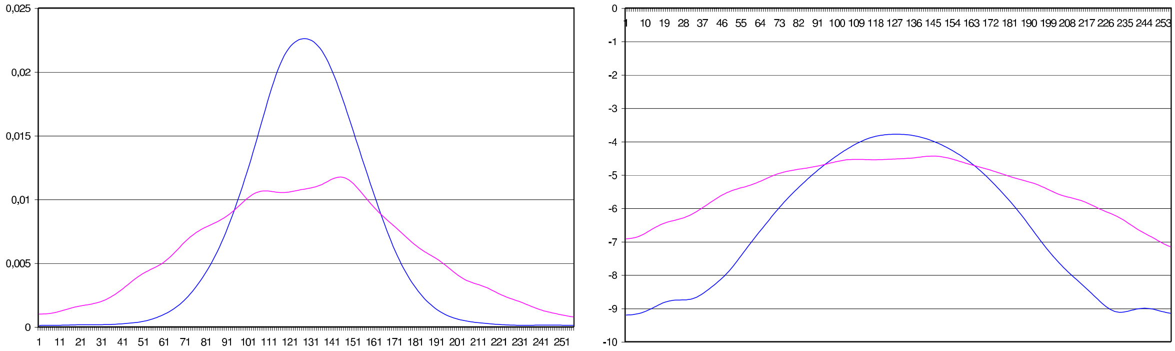


Fig. 5.10: Parzen density estimates for the image in Fig. 5.7 in the final state.
LEFT: BLUE LINE: p_1. MAGENTA LINE: p_2. RIGHT: BLUE LINE: $\log p_1$. MAGENTA LINE: $\log p_2$.

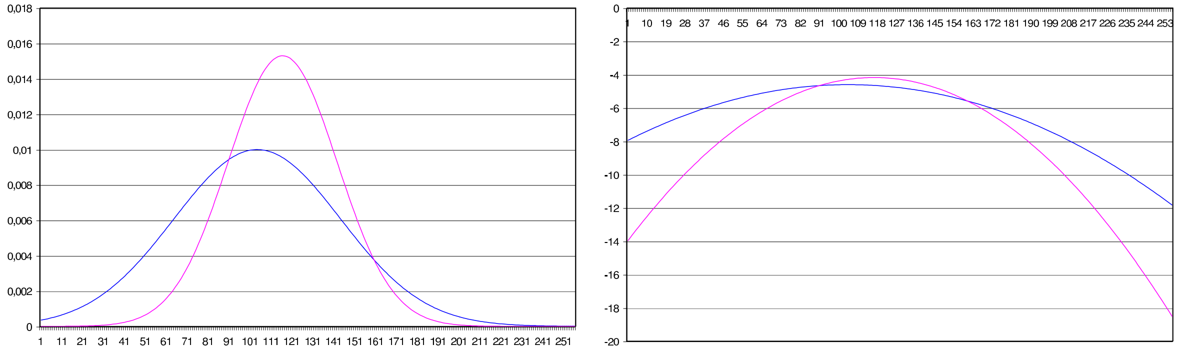


Fig. 5.12: Gaussian density estimates for the image and segmentation in Fig. 5.11.
LEFT: BLUE LINE: p_1. MAGENTA LINE: p_2. RIGHT: BLUE LINE: $\log p_1$. MAGENTA LINE: $\log p_2$.

This is different in Fig. 5.11. For the moment it is not important how the segmentation has been obtained. Fig. 5.12 shows for these regions the Gaussian density estimates in comparison to the probability densities obtained with the Parzen estimator in Fig. 5.13. Obviously, the nonparametric estimates capture more details of the region properties. This allows, for instance, a multi-cue texture segmentation method to assign the black area of the shadow to the background, although dark values actually fit better to the stripes of the zebra. This issue will become clearer in Section 5.2.

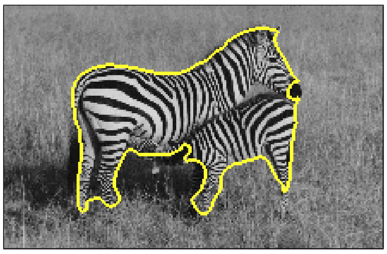


Fig. 5.11: Zebra segmentation.

There are some additional remarks concerning nonparametric density estimates:

- It is well-known in statistics that methods which capture many details run the risk to overfit the data. The important parameter in this respect is the standard deviation σ_h of the smoothing kernel. If it is chosen small, the estimated probability density fits the data very well, yet it does not take into account the uncertainty of the data, i.e. it is not reliable for ranges of values that rarely appear in the data. On the other hand, if σ_h is chosen large, the introduced uncertainty decreases the information content of the measurements. In the limit one obtains a uniform distribution, which is completely useless for segmentation.

- A good choice of σ_h basically depends on two factors: firstly on the amount of available data, and secondly on the model complexity one wants to choose for the segmentation. If there is only little data available, a Parzen density estimate with small σ_h becomes unreliable. Large σ_h are in this case necessary for a good approximation. In case of segmentation there are thousands of pixels available, nevertheless Fig. 5.10 shows that for the smaller region the underlying Gaussian distribution is not perfectly captured. Moreover, a complicated probability density function with many peaks provokes additional local minima in the energy functional. In the face of a good global solution one might thus be interested in reducing the complexity of the probability density by choosing σ_h comparatively large. In all further experiments $\sigma_h = 8$ is chosen.

- In practice one computes the Parzen density estimate by means of a discrete histogram. In this context it has to be noted that discretization introduces an additional (non-Gaussian) smoothing of the histogram.

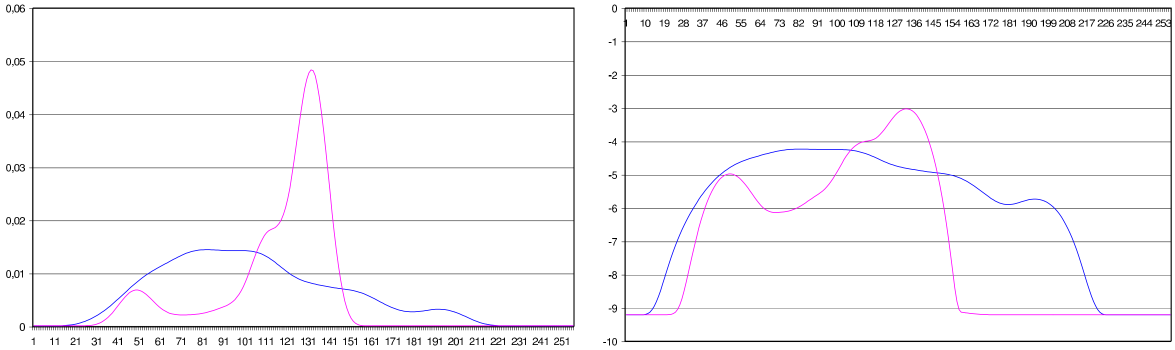


Fig. 5.13: Parzen density estimates for the image and segmentation in Fig. 5.11 ($\sigma_h = 8$). LEFT: BLUE LINE: p_1. MAGENTA LINE: p_2. RIGHT: BLUE LINE: $\log p_1$. MAGENTA LINE: $\log p_2$. For numerical reasons some small $\epsilon = 10^{-6}$ had to be added to p_i. Consequently, $\log p_i$ is limited from below.

Local region statistics. All region models discussed so far assume that the statistics do not change considerably within one region. Basically this is a reasonable assumption, since differences in the statistics indicate an object boundary. However, for some situations the model is too restrictive.

Consider, for instance, the blue sky of some (good) holiday picture. The sky is dark blue at the upper part of the region and almost white close to the horizon. The representation of the sky region by a single global probability density function is very inaccurate, as it contains only the information that the sky is blue. The information about which shade of blue can be expected at a certain position in the sky is lost. This prevents the distinction from another blue or white object, although locally there could be a significant difference in the regions' statistics.

Another example is the nose of the zebra in Fig. 5.11. Although locally it would better fit to the zebra region than to the background, the global statistics enforce the pixels to be assigned to the background[2]. This is because the dark pixels within the shadows almost outnumber the pixels in the black stripes of the zebra region. Globally regarded, this assignment is all right, since otherwise also the shadows would be assigned to the zebra. Locally, however, the assignment is not optimal.

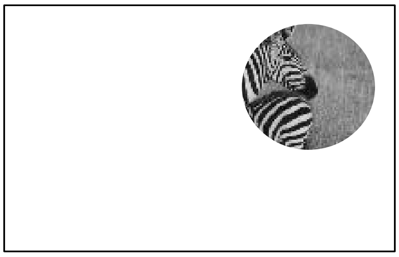


Fig. 5.14: Local cutout from Fig. 5.11.

The local information can be incorporated into the model, if the constant probability density function is replaced by a probability density function which can vary gradually within the region domain. That means, instead of the global probability density $p_i(s)$ for region i, one has to estimate a spatially varying probability density function $p_i(s, \mathbf{x})$. This can be achieved by computing at each location $\mathbf{x}$ a local probability density which employs a Gaussian window K_ρ centered at $\mathbf{x}$. If the window size ρ tends to infinity, the local estimates come down to the conventional global estimate.

[2] As one can see in Fig. 5.13, the nose should actually be assigned to the zebra region, yet the difference between both regions is so small that, with the chosen parameter setting, other feature channels prevent this.

There are the same possibilities how to model and estimate the local probability densities as for global densities:

1. *Gaussian probability density function with fixed constant standard deviation:*

$$p_i(s,\mathbf{x}) = \frac{1}{\sqrt{\pi}} e^{-(s-\mu_i(\mathbf{x}))^2} \qquad \mu_i(\mathbf{x}) = \frac{\int_{\Omega_i} K_\rho(\xi-\mathbf{x}) I(\xi)\, \mathbf{d}\xi}{\int_{\Omega_i} K_\rho(\xi-\mathbf{x})\, \mathbf{d}\xi} \tag{5.35}$$

   This model contains only one single parameter μ which has to be estimated. Thus the estimation is very reliable. On the other hand, the model is not well suited to distinguish regions with similar mean value.

2. *Gaussian probability density function:*

$$\begin{gathered} p_i(s,\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma_i(\mathbf{x})} e^{-\frac{(s-\mu_i(\mathbf{x}))^2}{2\sigma_i(\mathbf{x})^2}} \\ \mu_i(\mathbf{x}) = \frac{\int_{\Omega_i} K_\rho(\xi-\mathbf{x}) I(\xi)\, \mathbf{d}\xi}{\int_{\Omega_i} K_\rho(\xi-\mathbf{x})\, \mathbf{d}\xi} \qquad \sigma_i(\mathbf{x}) = \sqrt{\frac{\int_{\Omega_i} K_\rho(\xi-\mathbf{x})(I(\xi)-\mu(\mathbf{x}))^2\, \mathbf{d}\xi}{\int_{\Omega_i} K_\rho(\xi-\mathbf{x})\, \mathbf{d}\xi}} \end{gathered} \tag{5.36}$$

   Supplementing the standard deviation σ as a second free parameter tackles the problem of the previous model. The estimation of two parameters is still reliable, even if the size ρ of the window is rather small.

3. *Nonparametric Parzen density estimate:*

$$p_i(s,\mathbf{x}) = K_{\sigma_h,\rho} * \begin{cases} \mathbf{1}_{\{I(\mathbf{x})=s\}} & \text{if } \mathbf{x} \in \Omega_i \\ 0 & \text{else} \end{cases} \tag{5.37}$$

   where $K_{\sigma_h,\rho}(x,y,s) := K_\rho(x,y) K_{\sigma_h}(s)$ denotes an anisotropic Gaussian kernel with standard deviation σ_h in s-direction and standard deviation ρ along the x- and y-axis. The nonparametric model certainly offers the most precise description of the region statistics. However, the reliability of a Parzen density estimate depends very much on the available amount of data. Thus for small ρ this model is not appropriate.

Local region statistics have their advantages but also some drawbacks when compared to global statistics. Their main advantage is the capability to describe gradually varying regions. On the other hand, this also leads to a new problem: as gradual changes in the regions are allowed, one must now distinguish regions by sharp discontinuities in the statistics. This reminds of edge based active contours, which rely on discontinuities in the image itself. Indeed also local statistics evoke many local optima in the energy functional, in particular if ρ is small, and thus yield a high dependency on the curve's initialization.

Another disadvantage is the smaller amount of data available for estimating the local probability densities. While the whole region can contain thousands of pixels, there are fewer pixels within the local neighborhood. For this reason one should prefer the Gaussian model in order to estimate local densities. It is a reasonable tradeoff between model complexity and reliability.

A further problem with local statistics is their high computational effort. Whereas for the global statistics it is sufficient to compute for each region a single density estimate, for local statistics it is necessary to compute such a density at each location in the image, or at least within the narrow band.

Finally, it may also be considered as a drawback to have the size of the local window as an additional parameter ρ, though in comparison to global statistics it just offers additional choices beyond $\rho \to \infty$.

A continuation method for image segmentation. The problem of variational models for image segmentation yielding complex optimization problems with many local optima has been encountered already several times in this chapter. The number of local optima in general increases with a growing model complexity. This leads to a dilemma: on one hand, the model complexity is sought to be increased in order to accurately incorporate the image contents, but on the other hand a higher model complexity increases also the dependency of the result from the initialization. Thus an increased model complexity necessitates imperatively a strategy to cope with multi-modal optimization tasks.

In Section 2.3.3 continuation methods have been described as a strategy to heuristically prevent a method from getting stuck in a local optimum. Later on in Section 4.2.3, a continuation method has successfully been applied to the task of optic flow estimation. It is hence immediate to try to employ a continuation method also in the scope of image segmentation.

As described and illustrated in Section 2.3.3, continuation methods are based on the idea to start the optimization with smoothed, simplified versions of the energy functional and to successively increase the level of detail until finally a solution for the original problem is obtained. For image segmentation there are two possibilities how to create a smoothed version from the original functional:

1. Like in optic flow estimation, the input image can be regarded at coarser resolution levels. This removes details from the image which may disturb the method in finding the global optimum.

2. Additionally, the complexity of the region model can be reduced by replacing, for instance, a local density estimate by a global nonparametric density, or a nonparametric density by a Gaussian density.

Both approaches can as well be combined. This leads to the following coarse-to-fine strategy.

First a pyramid of successively downsampled versions of the input image is considered. Note that the multi-resolution optimization strategy has a positive side effect apart from the reduced dependency of the method from the initialization: it is considerably faster. At coarser resolution levels, the distance which the curve has to cover to reach its optimum position is much smaller than at the finest resolution level. Thus fewer iterations are necessary to converge to the steady state. Moreover, iterations at coarser levels are much faster, since there are fewer pixels. Consequently, the implementation with multiple resolution levels can be more than 10 times faster than a conventional implementation, depending on the image size; see Tab. 5.1 on page 133.

Additionally to the pyramidal approach, one can reduce the complexity of the region statistics at coarse levels. So one starts with a global Gaussian density estimate at the coarsest scale. At some finer scale, when the curve is already close to the object boundary, it is beneficial to switch to the nonparametric Parzen density estimate, since it can describe the region statistics more accurately than the Gaussian model. Also note that the amount of available data for the density estimate increases with finer levels. At the finest levels, one may finally want to switch to local statistics, as they yield the highest accuracy from all region models that have been discussed. An experimental evaluation of this coarse-to-fine strategy is covered by the experiments presented in the next section.

5.2 Multi-Cue Integration

The preceding section has introduced a basic model for two-region segmentation following the principles of geodesic active regions and the Chan-Vese model. The level set framework thereby provides a convenient representation of the region boundary, while the region statistics enforce an optimal splitting of the image according to its gray value distribution. A coarse-to-fine strategy approaches the problem of local optima, which gain influence with a growing model complexity.

This model is a good starting point, though images that are segmented satisfactorily by this model are rare and mainly artificial. It is obvious that the model has at least two important limitations:

- its restriction to exactly two regions in the image
- its limited capabilities to deal with textured images

Section 5.3 will deal with the first issue, whereas in this section the focus lies on the integration of additional information such as color, texture, and motion. In particular, the texture and motion features extracted by the methods introduced in Chapter 4 are incorporated into the model.

Model for multiple features. For this purpose, the model from the last section has to be extended from gray scale images I to feature vector images $\mathbf{F} = (F_1, ..., F_M)$. This extension is actually very straightforward and has been proposed for geodesic active regions [PD02b] as well as the Chan-Vese model [CSV00]. In both cases it is assumed that the channels F_j are independent, thus the total a-posteriori probability density $p_i(\mathbf{F})$ of region Ω_i is the product of the separate probability densities $p_{i,j}(F_j)$:

$$p_i(\mathbf{F}) = \prod_{j=1}^{M} p_{i,j}(F_j). \tag{5.38}$$

Thus the energy functional for segmentation with multiple feature channels reads:

$$E(\Phi) = \int_{\Omega} \left(-H(\Phi) \sum_{j=1}^{M} \log p_{1,j} \; - (1 - H(\Phi)) \sum_{j=1}^{M} \log p_{2,j} \; + \nu |\nabla H(\Phi)| \right) \mathbf{dx} \tag{5.39}$$

and yields the following gradient descent equation:

$$\partial_t \Phi = H'(\Phi) \left(\sum_{j=1}^{M} (\log p_{1,j} - \log p_{2,j}) + \nu \operatorname{div} \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) \right). \tag{5.40}$$

Due to their independence, the $p_{i,j}$ can be estimated for each region i and channel j separately. Hence, the statistical models introduced in the last section can be applied analogously.
In cases where the correlation between channels is important for discrimination one can either estimate multi-dimensional probability densities, e.g., by a multi-dimensional Gaussian as proposed in [RD03], or alternatively by providing additional channels with the correlation of features. Since in particular nonparametric density estimates entail a very high computational effort when extended to higher dimensions, the latter concept can be advantageous.

Possible features to be incorporated by means of this model are color, texture, and motion cues. In the case of color, it is reasonable to transform the RGB color space to CIELAB, in which the Euclidian distance measure is supposed to better reflect human perception [Mac42, Int71].

Concerning texture, one can often find Gabor features in connection with image segmentation, see e.g. [SSZ01, PD02a][3]. In Section 4.1 a sparse alternative to the Gabor feature space has been suggested. As Gabor features it discriminates textures by means of their magnitude, orientation, and scale. However, the sparse representation reduces the high redundancy of Gabor features. Note that for the model in (5.39), a probability density needs to be estimated for each feature channel. Thus the reduction from 12 to 4 texture features can yield a speedup of up to factor 3. As soon as correlations between these features may be taken into account, either by additional feature channels or multi-dimensional probability density estimates, this speedup can get even larger. Later on, the texture segmentation quality with Gabor features and the sparse feature space will be compared in an experiment.

Besides color and texture, motion cues can be good indicators for separating objects. With the optic flow model introduced in Section 4.2.3 a quite reliable estimate for the optic flow vector (u, v) can be computed. Incorporation of u and v into the feature vector is supposed to improve the segmentation of moving objects.
Motion segmentation has been investigated in several alternative approaches, e.g. in [Pot75, Tho80, BF93, PD00b, MP02, CS05, PD05]. In many of these approaches, not the optic flow but simply the difference between successive images is used as a cue for segmentation. Other works do not use the optic flow as a pre-computed feature, but estimate it simultaneously with the segmentation [MP02, CS05, PD05]. This is principally favorable, as the evolving segmentation can improve the motion estimates. However, it is not trivial to connect motion estimation and segmentation concepts, especially when elaborated models of both research fields are to be included. While optic flow is incorporated here as a pre-computed feature, it is a topic of further research to combine the optic flow method as presented in Section 4.2.3 with a level set based motion segmentation approach, such as in [CS05, PD05].

Coupled nonlinear diffusion of the feature space. When the sparse texture feature space has been derived in Section 4.1.3, it has been proposed to apply a coupled nonlinear diffusion process. The main motivation for this nonlinear diffusion has been the nonlinear structure tensor which employs nonlinear diffusion for the local integration of structure information.
However, the coupled nonlinear diffusion process can as well be motivated as a supplement to the coarse-to-fine strategy which reduces the risk of the segmentation method to get stuck in local minima. Applying coupled nonlinear diffusion to the feature vector $\mathbf{F}$ at each resolution level, adds some additional smoothing, while keeping important image edges. With edge enhancing flow[4], joint discontinuities in the features are even enhanced, what can improve the interaction of feature channels in the segmentation.
Thus it is proposed to restrict the smoothing not only to the texture features but to smooth the whole feature vector $\mathbf{F}$. At each resolution level, nonlinear diffusion according to the vector-valued diffusion scheme in (2.7) and edge enhancing flow with $p = 1.5$ is applied to $\mathbf{F}$. In the experiments, the influence of this nonlinear diffusion process will become evident.

[3] See also Section 4.1.

[4] See Section 2.1.2.

A note on feature weighting. A model that uses several different unrelated features gives rise to the question how these features are to be weighted against each other. This problem of comparing apples and oranges has already been faced in Section 4.1.3, when the components of the structure tensor had to be combined with the local scale measure. As the potential range of these features has been limited, a pragmatic procedure has been the scaling of the features to the same range as the image gray value.

Analyzing the influence of feature weighting on the segmentation leads to an interesting observation. While comparable ranges of all features are important for coupled nonlinear diffusion to work properly, the weighting of each channel has only little or even no influence on the discrimination properties of the curve evolution, if appropriate statistical models are employed. Consider the Gaussian probability density estimate. By multiplying a feature with a factor a, i.e., increasing its contrast by this factor, both the standard deviation σ_i and the difference of each value to the mean μ_i are scaled by a. For the region term $\log p_1 - \log p_2$ in the Euler-Lagrange equation this yields:

$$\begin{aligned} & -\log(\sqrt{2\pi}\, a\, \sigma_1) - \tfrac{(a(s-\mu_1))^2}{2(a\,\sigma_1)^2} + \log(\sqrt{2\pi}\, a\, \sigma_2) + \tfrac{(a(s-\mu_2))^2}{2(a\,\sigma_2)^2} \\ = \; & -\log(\sigma_1) - \tfrac{(s-\mu_1)^2}{2\sigma_1^2} + \log(\sigma_2) + \tfrac{(s-\mu_2)^2}{2\sigma_2^2}. \end{aligned} \tag{5.41}$$

This means, the region force is independent from a, and therefore contrast independent. Instead, the influence of a feature depends on how well it distinguishes the two regions, i.e. its discriminative power.

One can easily assure oneself that, apart from a linear scaling of the smoothing parameter σ_h, also the Parzen density estimate is contrast invariant. Only the simplified Gaussian model with a fixed standard deviation, which is used in the Chan-Vese model, depends crucially on the relative contrast of the features. This is another good reason for applying at least the full Gaussian model to describe the interior of regions.

Parameter setting. When the segmentation model has been derived in this chapter, one may have got the impression that it contains many parameters that are difficult to choose. In fact, such a large set of critical parameters would be a severe shortcoming of the method, as it is supposed to deal with *unsupervised* segmentation, whereas the setting of a bunch of parameters is certainly a supervised action.

However, the number of relevant parameters is quite restricted. Most parameters just appeared due to more general models, i.e., the simplified model implicitly contains the parameter as well[5]. Apart from two exceptions, where the choice depends a lot on the image size, it is therefore relatively easy to set the parameters to a fixed value. In fact, the experiments will show that the method is capable to successfully process 30 rather different images with a single(!) fixed set of parameters.

[5] Recall, e.g., from Section 2.2.2 the nonlinear structure tensor which is a generalization of the conventional structure tensor. Actually both structure tensors contain the parameter p that steers the amount of edge preservation, only in the conventional structure tensor, p is fixed to 0, which means there is no edge preservation. This example clearly shows that it is not necessarily a drawback to have the possibility to set a parameter to another value.

In the following, all parameters and their fixed values are made explicit:

- Starting with the extraction of the texture features, there is only the stopping time T used in the scale measure. Under the assumption that the image contains values in $[0, 255]$, it can be easily set to a fixed value $T = 20$. If there is really a very large variety of texture scales to be extracted, one could still add a second version of the scale measure with another T to the features space, yet this is not recommended here.
- Nonlinear diffusion applied to the feature space entails two further parameters: the amount of edge preservation p and the diffusion time. The choice of p is uncritical. It is set to $p = 1.5$. The diffusion time has more influence on the result: if there is too much diffusion, there is too much information lost for the statistics. On the other hand, if there is not enough diffusion, the risk to hit a local optimum increases. Furthermore, it is not easy to find an appropriate fixed value for this parameter, since the optimum choice depends very much on the size of the image, the number of feature channels, and the present amount of texture.
  This parameters is therefore replaced by another parameter which is easier to choose. In particular, the diffusion process is stopped as soon as the average total variation per pixel and channel falls below a certain threshold ϑ. This parameter is by construction independent from the image size and the number of feature channels. Moreover, it takes into account that in case of heavy texture there is more diffusion necessary. Consequently, this surrogate parameter can be set to the fixed value $\vartheta = 6$. In combination with the coarse-to-fine strategy, ϑ is linearly increased at each level such that there is no diffusion at the finest level anymore.
- The curve evolution introduces only the weighting parameter ν for the length constraint. Again the optimum choice of this parameter depends a lot on the image size. In experiments it has turned out that the optimum ν grows faster than the perimeter of the image, but slower than its area $|\Omega|$. Thus, the parameter is set to $\nu = 0.001|\Omega|^{0.7}$. One can see later in the experiments that this adaptive choice leads to good results for a large bandwidth of image sizes.
- If nonparametric density estimates are used for the regions statistics, the standard deviation σ_h of the Gaussian kernel appears as a free parameter. Its choice has already been discussed in the last section. For the experiments it has been set to $\sigma_n = 8$.
  If the nonparametric densities are replaced by global Gaussian densities, there is no such parameter. In case of local Gaussian densities, there appears another parameter instead, that is the standard deviation ρ of the Gaussian used to determine the local window. It has to be chosen large enough to obtain reliable density estimates. In the experiments it has been set to $\rho = 16$.
- Besides these model parameters there are some numerical parameters. First there is the downsampling factor of the coarse-to-fine strategy, which has been set to $\eta = 0.5$. Further on there is the number of iterations. At the coarsest level the initialization with horizontal stripes is generally very far from the optimum solution. Hence, 300 iterations have been conducted. At the finer levels, the curve initialization from the previous level is already quite good, thus the number of iterations has been reduced to 50. For the local statistics only 10 iterations have been performed.

Experiments on texture segmentation. In order to evaluate the segmentation method and the influence of the different techniques involved, an extensive experiment on 30 textured test images has been performed. The images are depicted in Fig. 5.16 - Fig. 5.44, some of them being synthetic and some being real-world images. They have been chosen such that there is mainly one object region in front of a more or less homogeneously textured background. This takes the model assumptions and restrictions of the method into account: the image domain is always split into two regions, and there is no prior knowledge about the appearance regions, besides the assumption that the data is homogenous inside a region and inhomogeneous between regions.

In order to evaluate the influence of certain parts of the technique, the tests have been performed with six different versions of the method:

(a) *Proposed method with global statistics at all scales.*
In this setting, the method yields nice results for all 30 test images. Only the head of the leopard in Fig. 5.30 is not captured correctly.

(b) *Local statistics at the finest scale.*
Switching at the finest level to local statistics allows the method to be slightly more accurate in some cases. Examples are the baby tapir in Fig. 5.35, the tail of the tiger in Fig. 5.37, or the neck of the zebra in Fig. 5.44. Also most parts of the leopard head in Fig. 5.30 can be captured. In the other images the results are very similar to (a). While this is the best performing method, one may wonder if it is worth the additional computational effort documented in Tab. 5.1. Local statistics will become more important in Sec. 5.4 when image-driven segmentation is combined with prior shape knowledge.

(c) *Gabor features instead of the proposed texture features.*
It is a legitimate question how the sparse texture feature space compares to the popular Gabor feature space, which is well-established and very easy to implement. From Tab. 5.1 one can see the expected advantage of the sparse representation: it leads to a speed-up factor of 2. Furthermore, the Gabor features fail in case of the tapir in Fig. 5.35 and the fish in Fig. 5.28. Also the result for the zebra in Fig. 5.39 is considerably worse than in (a). In all other cases there are only minor differences. This shows that in a few cases the sparse representation can even help to discriminate textures while there is no example among the test images where it performs considerably worse than the Gabor features.
Note that in this setting the parameter ν has been increased by factor 2 in order to yield optimum results for all images.

(d) *No nonlinear diffusion applied to the feature space.*
Nonlinear diffusion applied to the feature space is basically some kind of pre-segmentation. Its purpose is to help the segmentation method to find the optimum solution by simplifying the data without destroying too much of the important contents. However, the question may arise whether this pre-segmentation part is actually necessary.
Indeed, the results of (a) and (d) are very similar in most cases. Only in Fig. 5.23, Fig. 5.34, and Fig. 5.35 there are some significant differences preferring (a). Interestingly, these are among the most difficult test images. So it seems that nonlinear diffusion can help the method in difficult scenarios to find a more attractive optimum.

(e) *No coarse-to-fine optimization.*
While deviations from (a) so far did not reveal revolutionary different results, differences become much more significant if the coarse-to-fine strategy is not applied anymore. Firstly, Tab. 5.1 reveals the sharp increase in computation time, secondly most results get worse or even fail completely. Note that there is still nonlinear diffusion with $\vartheta = 6$ applied to the features, otherwise the method would fail in even more cases. On the other hand, the relatively large amount of diffusion also destroys a lot of information

needed for the statistical model. This explains why the segmentation fails in Fig. 5.15, Fig. 5.28, and Fig. 5.43, although these are rather simple test images. By removing also the diffusion part one obtains better segmentations for these images, but in return the method fails in many other cases; see Fig. 5.46.

An interesting observation is the good segmentation of the leopard in Fig. 5.30 including the head. This is because (e) considers only a single scale, at which the Gaussian probability density estimate is applied. In this specific case, the Gaussian estimate is favorable to capture the leopard's head.

In this setting, again the parameter ν has been multiplied by 2 to yield optimum results.

(f) *Without any texture features.*

When turning off the texture features, the segmentation of most test images fails. This is not too surprising. On the other hand, one may be astonished that for some images the segmentation does *not* fail, although there is heavy texture in these images, see e.g. Fig. 5.38. The explanation is given by the statistical model, which describes the region not only by its mean, but also by its standard deviation, or even a nonparametric probability density function. Thus, provided a good initialization, all textures that can be distinguished reliably by the gray value histogram can also be distinguished by the segmentation method, even when explicit texture features are missing. This can be important even when texture features are supplemented, because it avoids situations where feature channels contradict each other.

Fig. 5.45 illustrates the curve evolution at the distinct levels of the coarse-to-fine strategy for test case (a) and the squirrel image. As in all the other experiments, the curve has been initialized with 8 horizontal stripes. One can see that the curve captures the coarse shape of the squirrel very quickly. Obviously the important task of detecting the object is already performed at the coarsest scale. This is exactly what has been intended by the coarse-to-fine strategy: the coarsest scale has to find a good initialization, which leads the curve safely to a satisfactory optimum. It is therefore the critical point of the whole segmentation procedure.

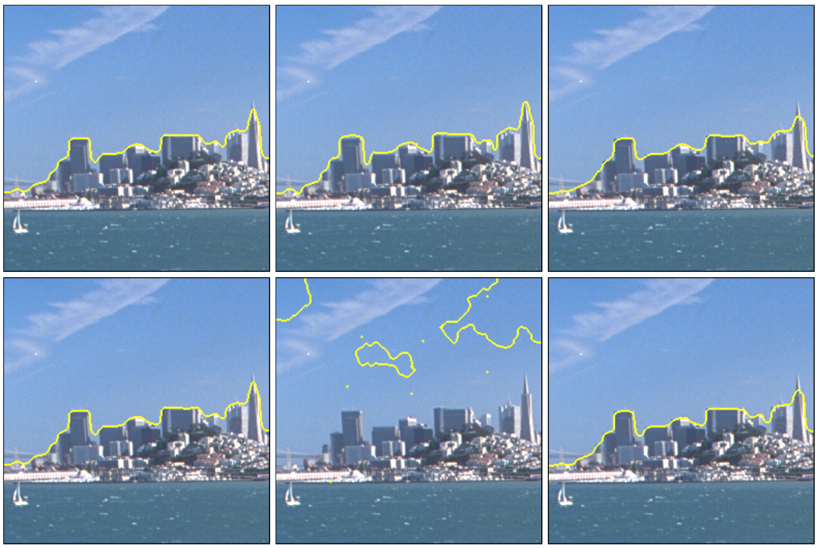


Fig. 5.15: FROM LEFT TO RIGHT, TOP TO BOTTOM:
(a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features.
(d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

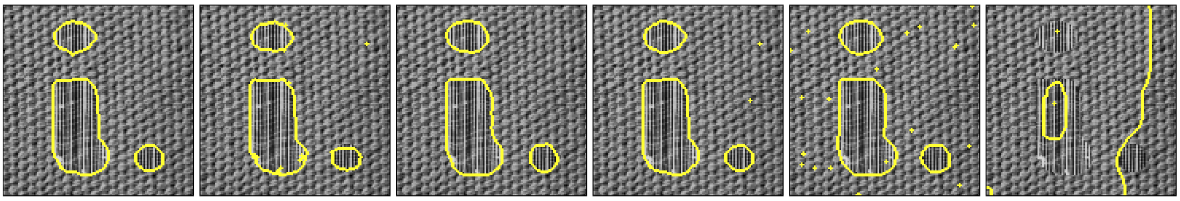


Fig. 5.16: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

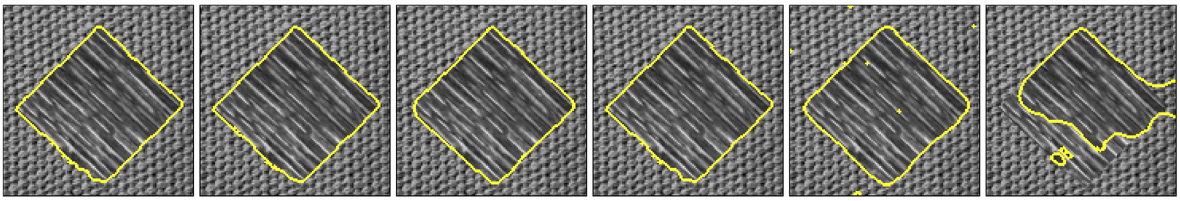


Fig. 5.17: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

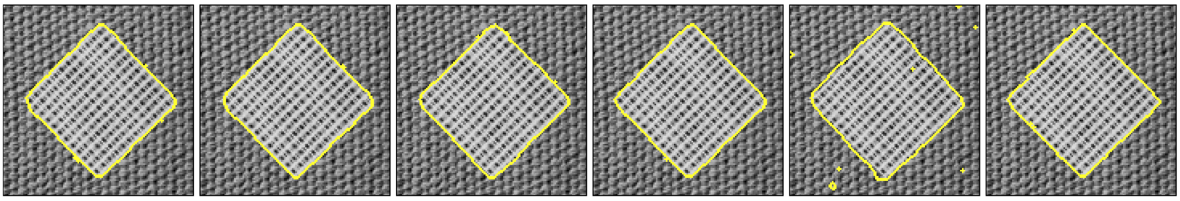


Fig. 5.18: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

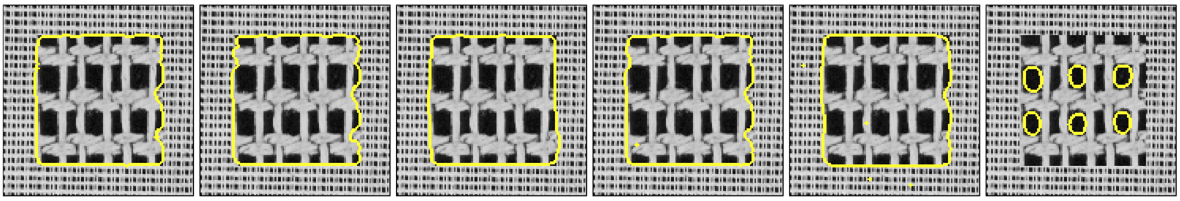


Fig. 5.19: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

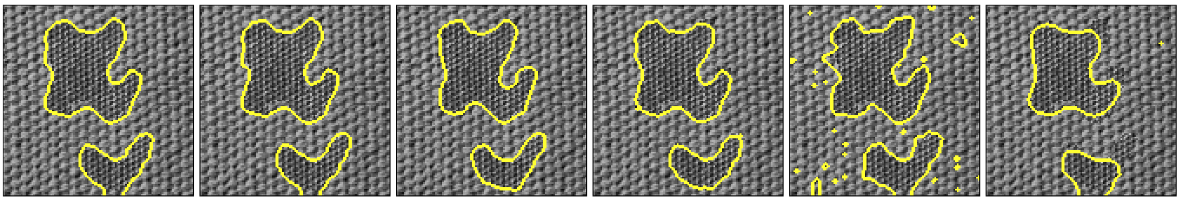


Fig. 5.20: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

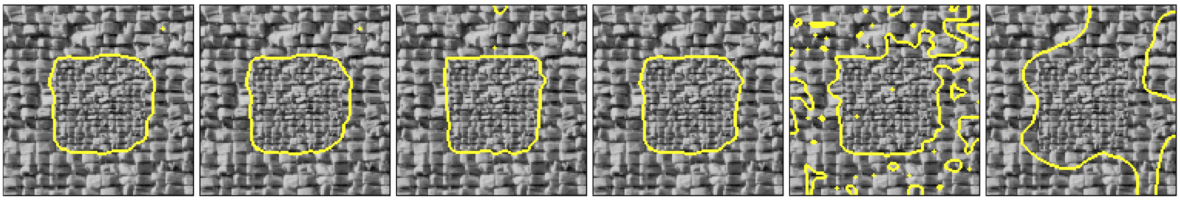


Fig. 5.21: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

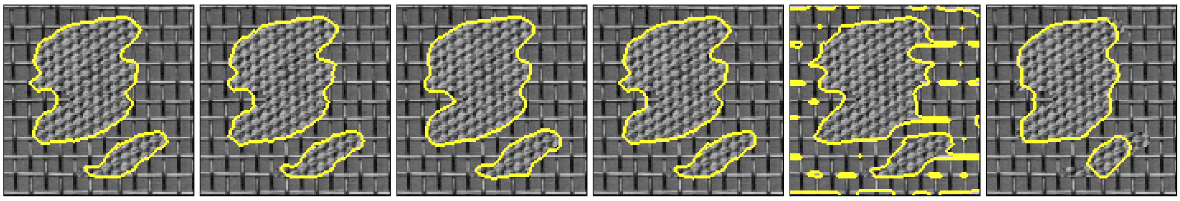


Fig. 5.22: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

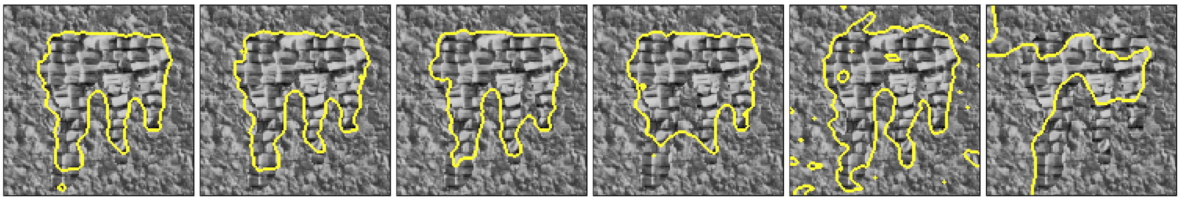


Fig. 5.23: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

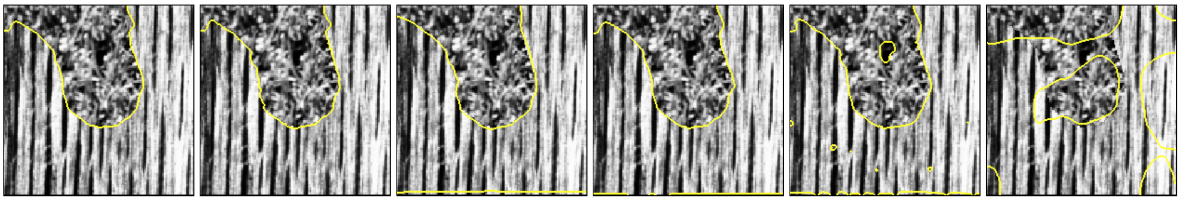


Fig. 5.24: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

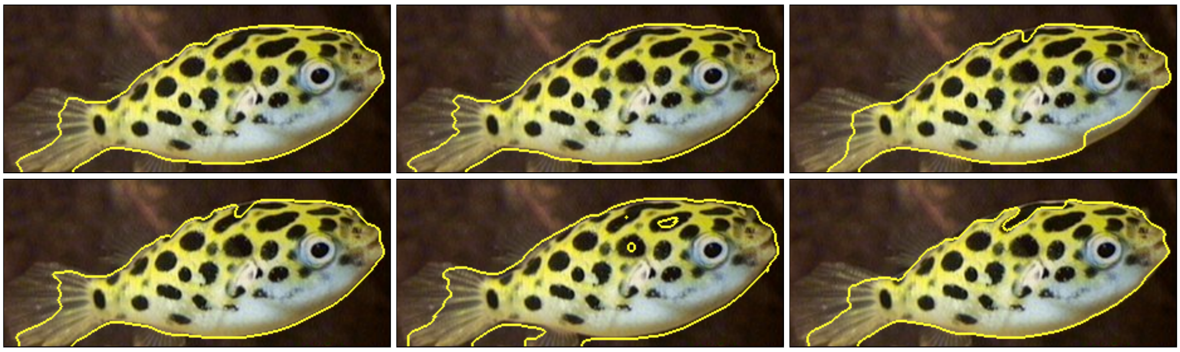


Fig. 5.25: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

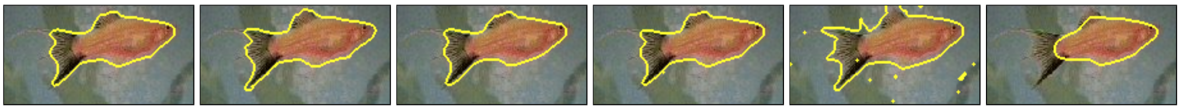


Fig. 5.26: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

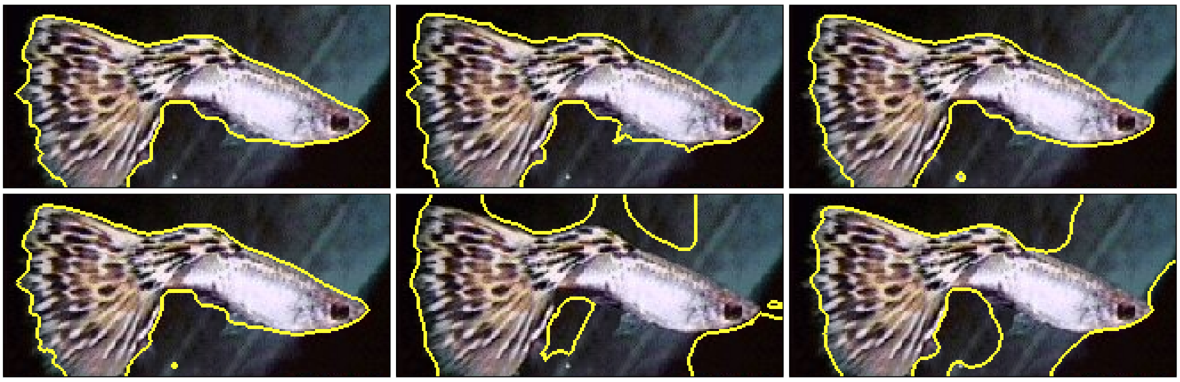


Fig. 5.27: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

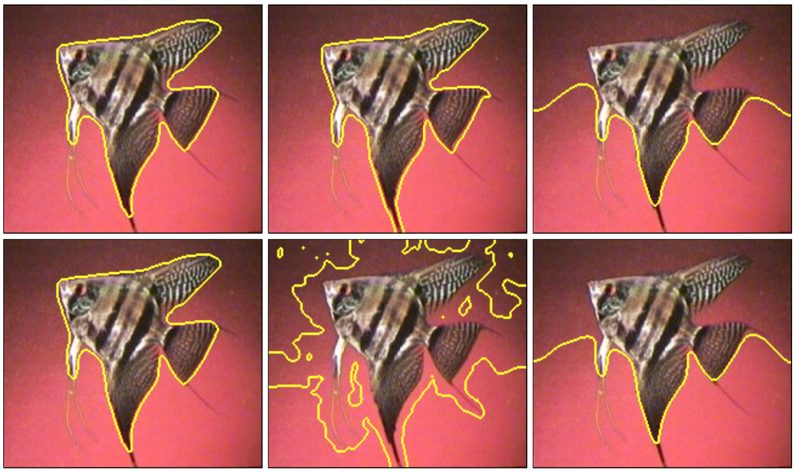


Fig. 5.28: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

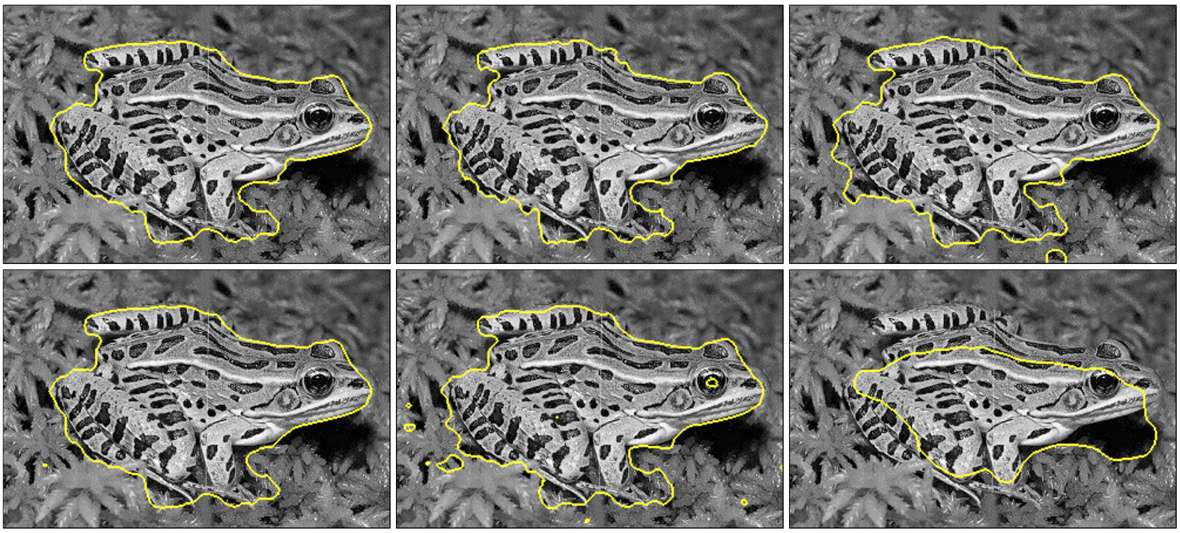


Fig. 5.29: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

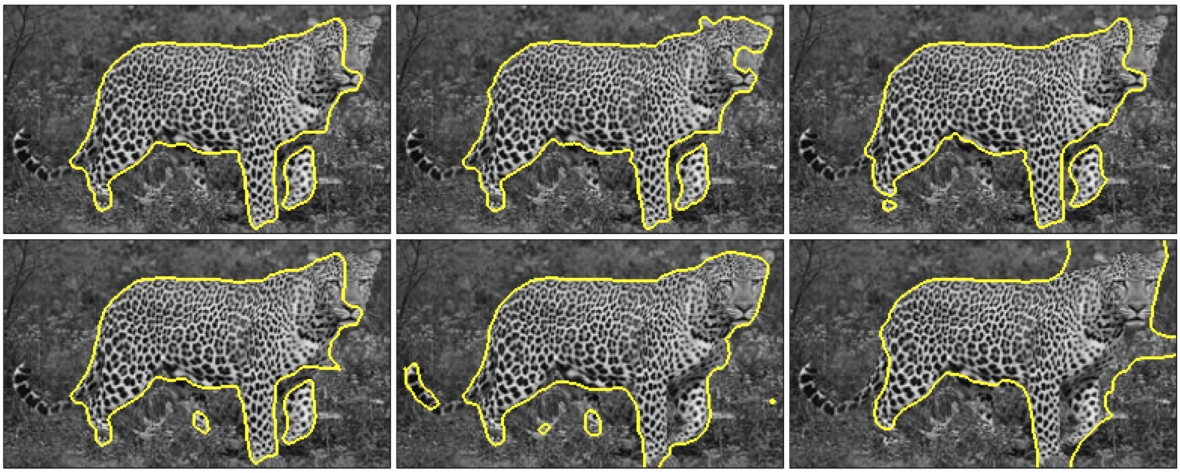


Fig. 5.30: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

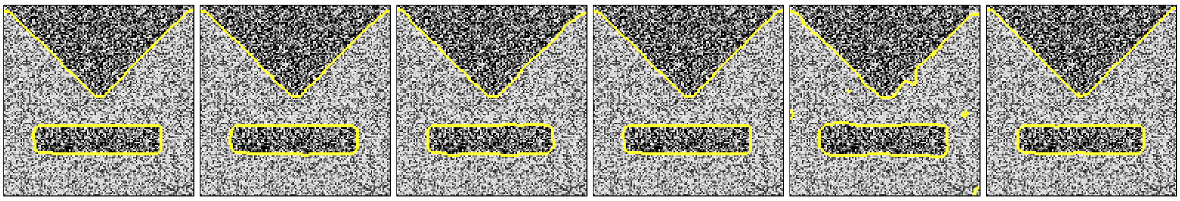


Fig. 5.31: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

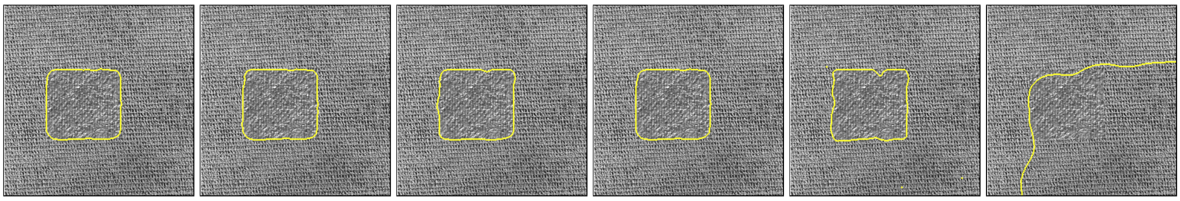


Fig. 5.32: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

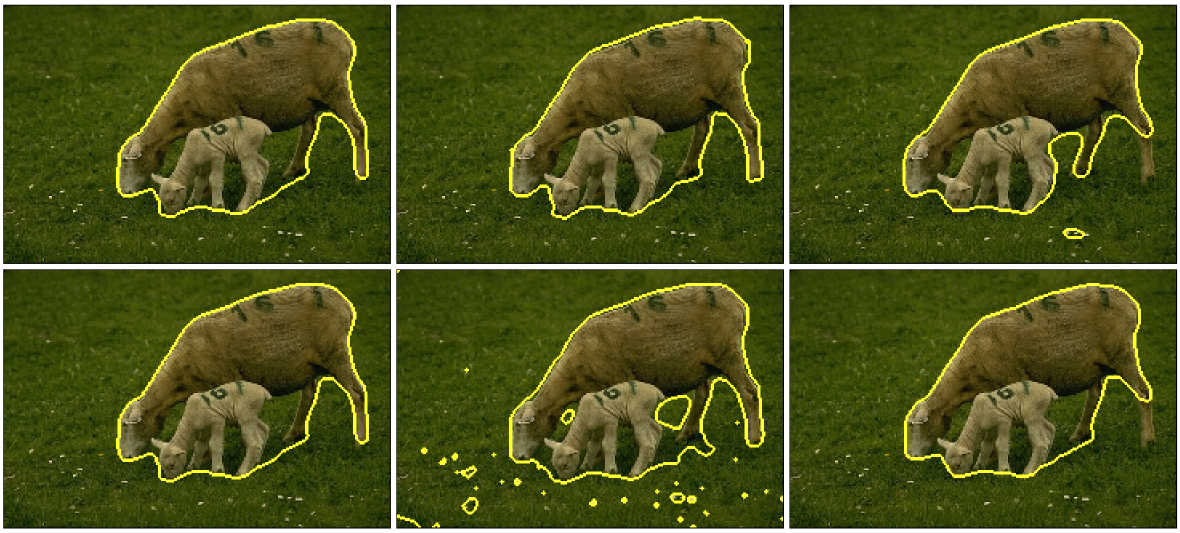


Fig. 5.33: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

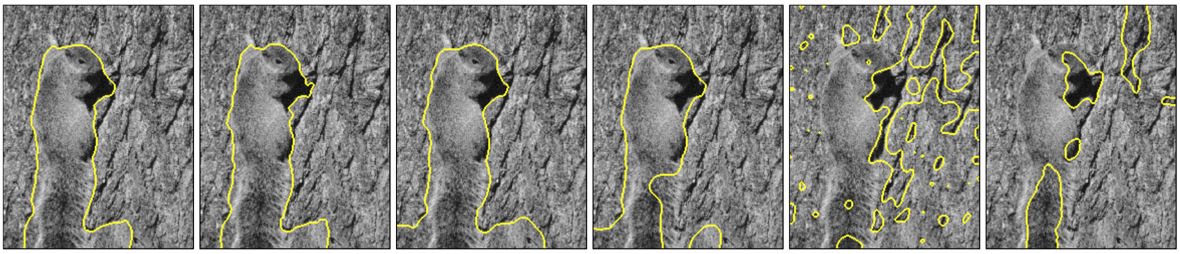


Fig. 5.34: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

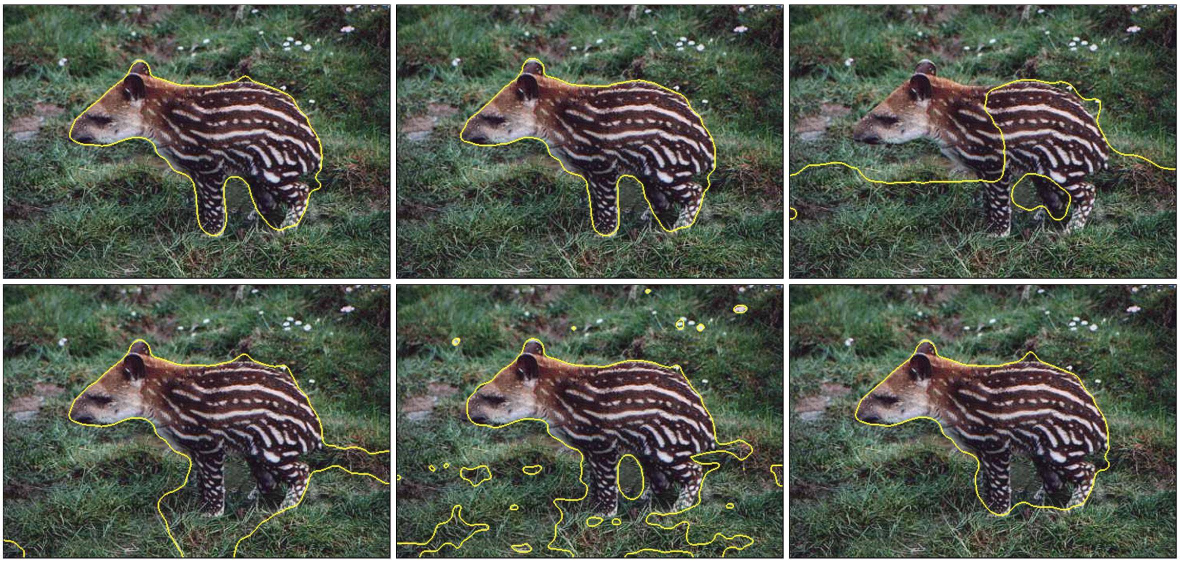


Fig. 5.35: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

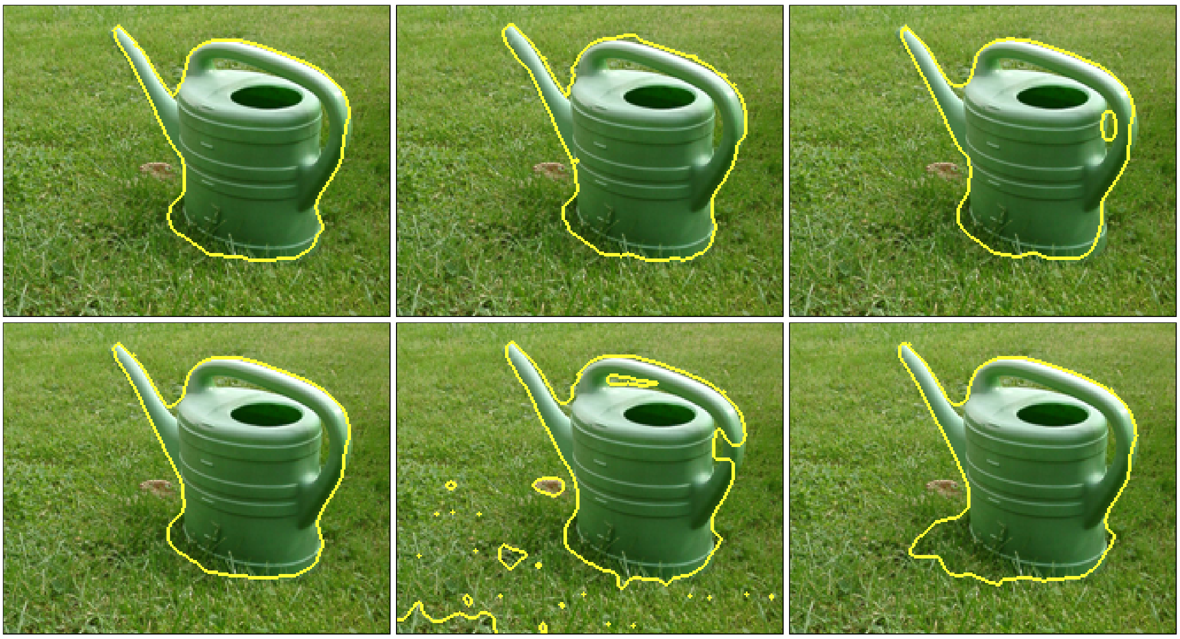


Fig. 5.36: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

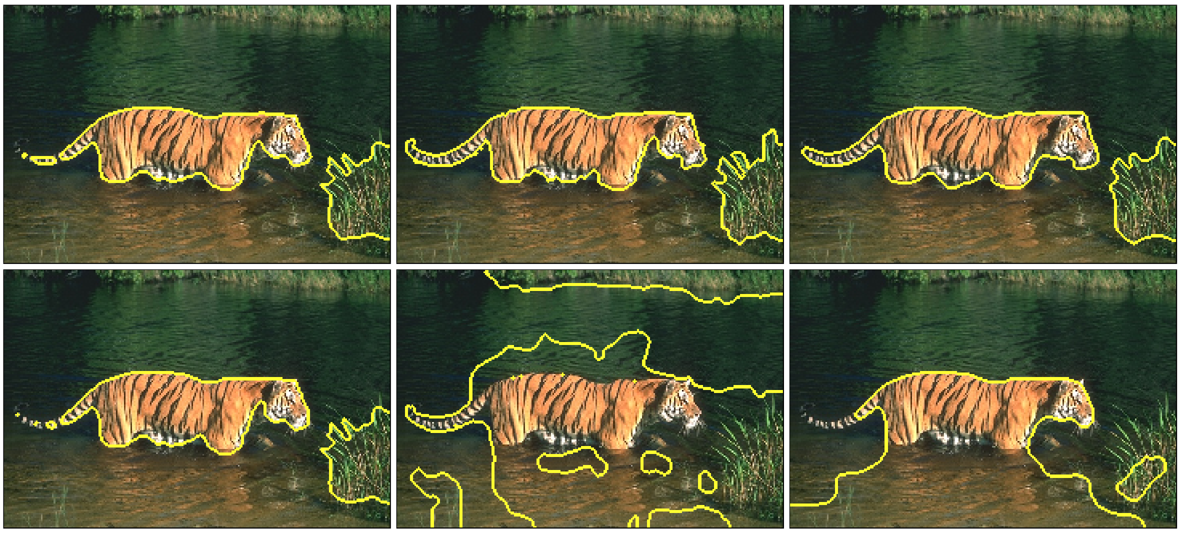


Fig. 5.37: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

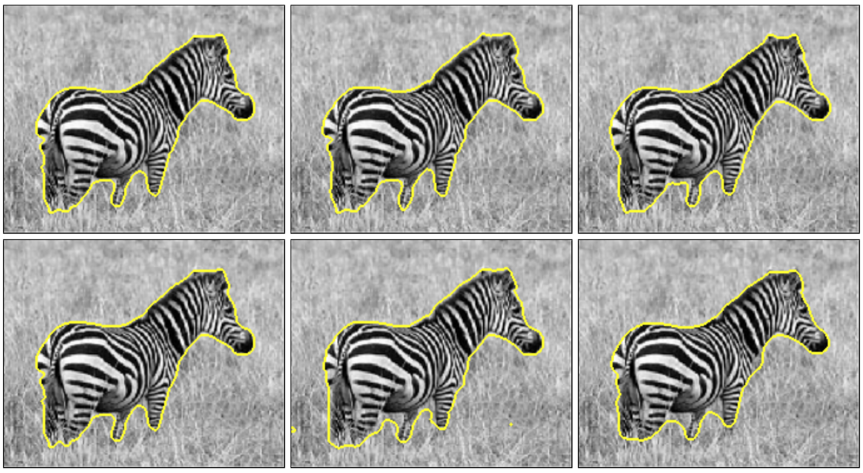


Fig. 5.38: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

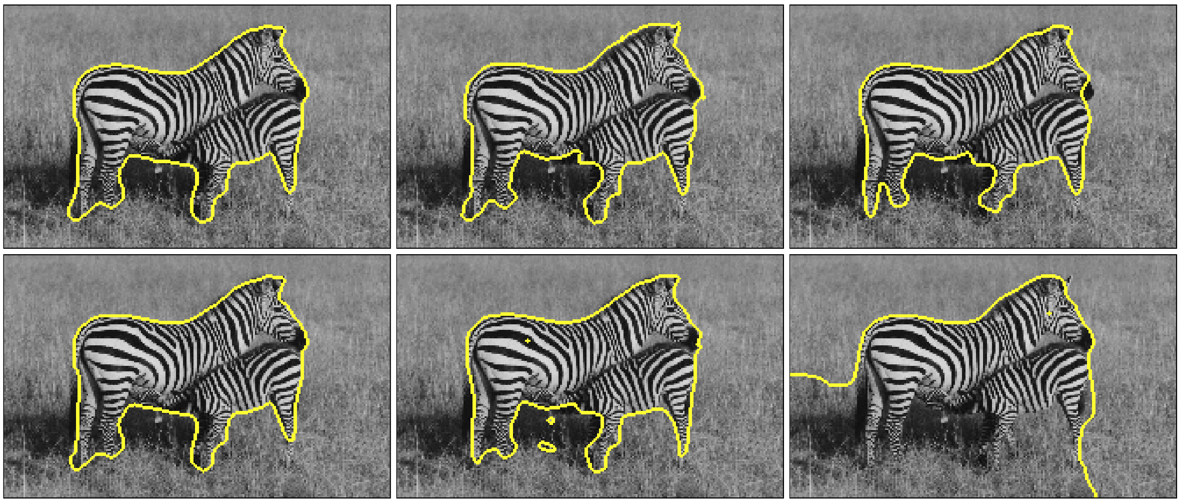


Fig. 5.39: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

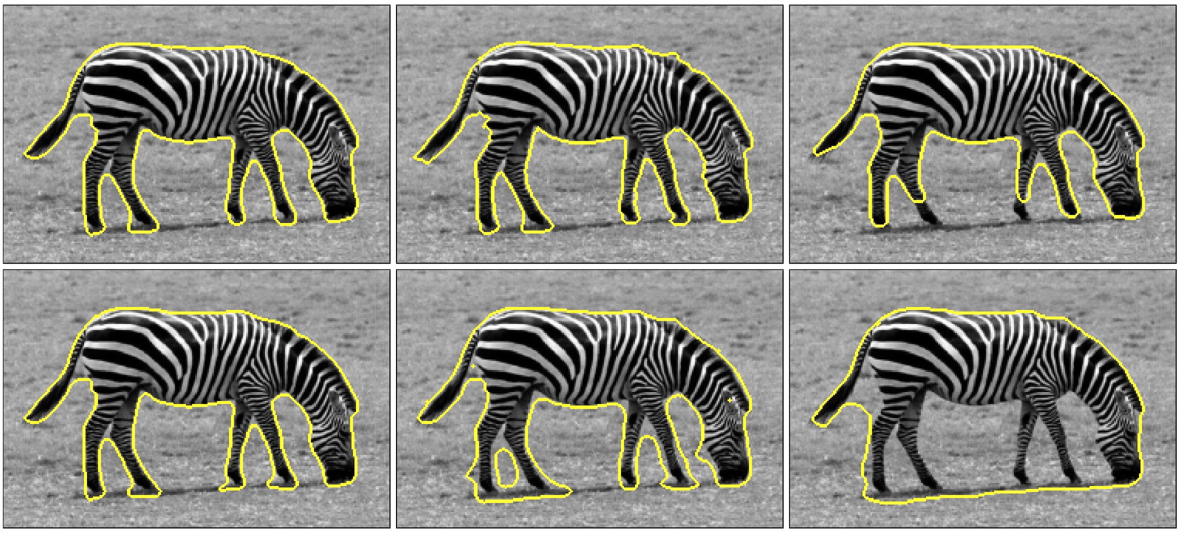


Fig. 5.40: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

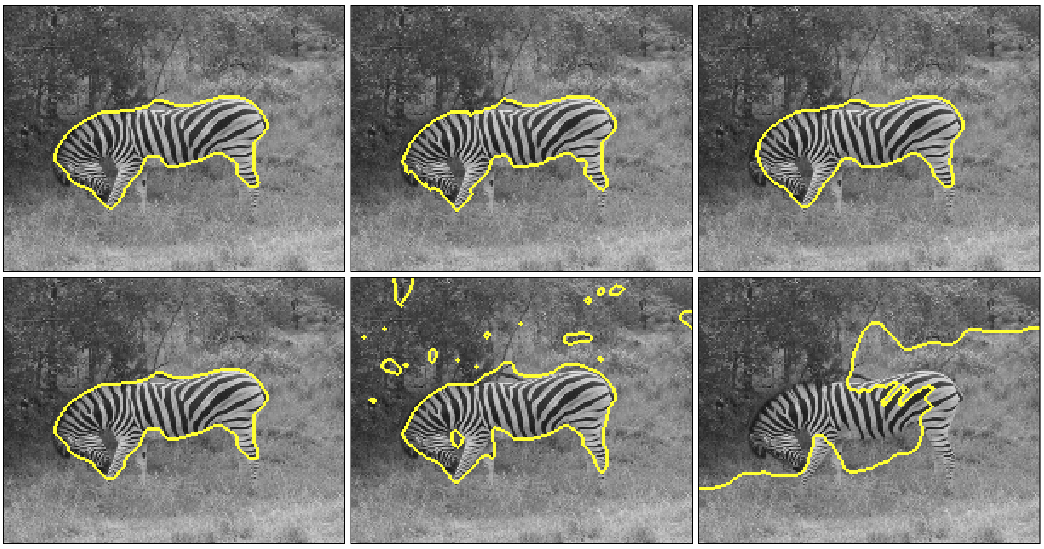


Fig. 5.41: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

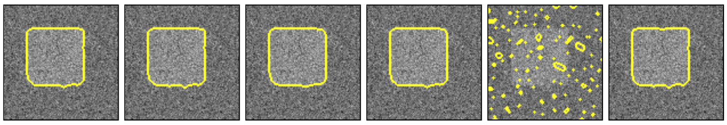


Fig. 5.42: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

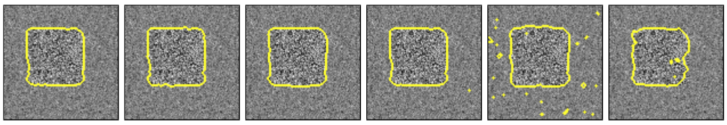


Fig. 5.43: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

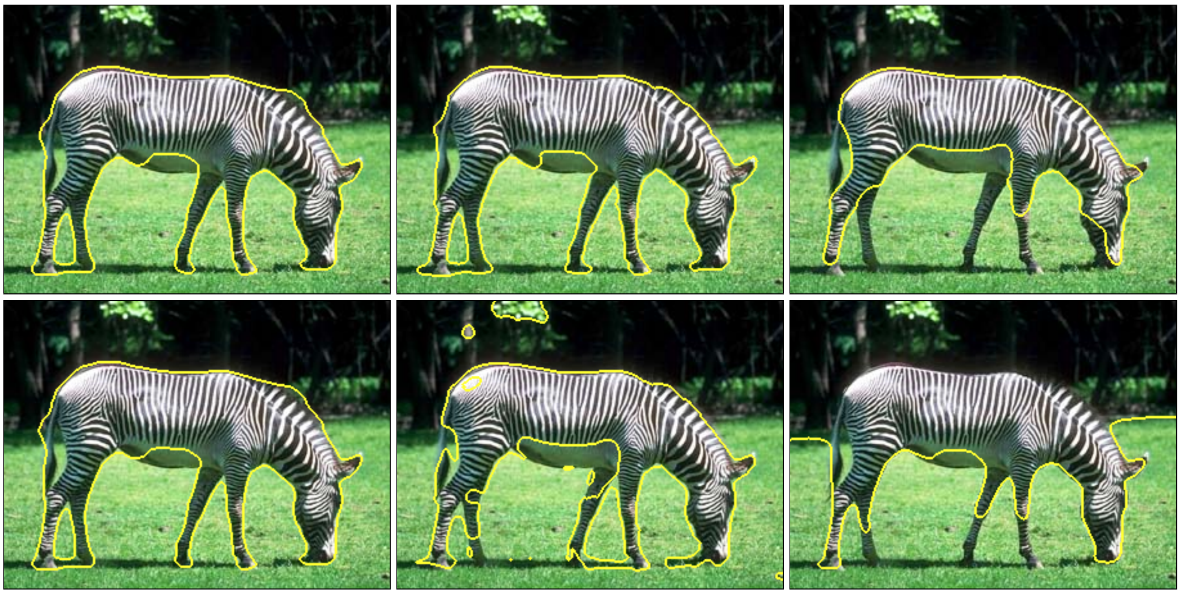


Fig. 5.44: (a) Global statistics. (b) Local statistics at finest scale. (c) Gabor features. (d) Without diffusion of feature space. (e) Without multi-scale. (f) Without any texture features.

Image	Size	Without local statistics	With local statistics	Gabor features	Without diffusion	Without multi-scale	Without texture
Fig. 5.15	256×256	9.8	42.8	20.7	9.0	115.0	5.2
Fig. 5.16	119×121	1.7	16.3	3.5	1.6	16.7	.80
Fig. 5.17	119×123	1.7	12.4	3.4	1.5	17.1	.74
Fig. 5.18	118×118	1.6	11.8	3.2	1.5	16.8	.71
Fig. 5.19	120×122	1.7	19.1	3.4	1.5	18.1	.74
Fig. 5.20	109×113	1.5	18.4	3.1	1.4	15.6	.63
Fig. 5.21	123×119	1.7	14.4	3.5	1.6	21.7	.79
Fig. 5.22	115×113	1.6	20.0	3.3	1.5	19.4	.70
Fig. 5.23	121×122	1.7	23.6	3.4	1.5	19.6	.74
Fig. 5.24	204×204	4.5	21.6	9.2	4.2	62.7	2.5
Fig. 5.25	307×131	5.7	47.7	12.2	5.4	67.4	3.1
Fig. 5.26	148×75	1.7	19.9	3.7	1.6	15.3	.83
Fig. 5.27	217×100	3.3	36.1	7.2	3.2	32.7	1.7
Fig. 5.28	212×184	5.5	52.6	11.5	5.3	71.0	2.8
Fig. 5.29	329×220	8.6	53.9	16.9	7.8	133.0	4.5
Fig. 5.30	256×151	4.3	45.1	8.8	4.0	60.0	2.1
Fig. 5.31	128×128	1.9	20.0	4.0	1.7	22.1	.84
Fig. 5.32	255×246	6.9	27.3	13.4	6.4	106.6	3.7
Fig. 5.33	241×161	5.5	42.8	11.8	5.2	67.8	2.9
Fig. 5.34	189×244	5.1	33.6	10.2	4.8	83.9	2.6
Fig. 5.35	489×346	27.9	105.0	56.7	26.7	459.5	17.2
Fig. 5.36	218×178	5.4	42.1	11.7	5.1	66.0	2.9
Fig. 5.37	241×161	5.7	56.2	12.0	5.4	69.7	3.1
Fig. 5.38	224×177	4.4	35.6	8.7	4.0	60.0	2.1
Fig. 5.39	220×140	3.3	32.2	6.8	3.1	42.6	1.5
Fig. 5.40	250×167	4.8	61.2	9.6	4.5	67.0	2.3
Fig. 5.41	200×160	3.2	21.8	6.7	3.0	43.2	1.6
Fig. 5.42	100×100	1.1	11.0	2.4	1.1	11.6	.50
Fig. 5.43	100×100	1.2	11.5	2.4	1.1	11.3	.50
Fig. 5.44	300×225	10.4	110.0	21.3	9.9	147.0	5.5

Tab. 5.1: Computation times in seconds on an AMD Athlon XP1800+ (C++ implementation).

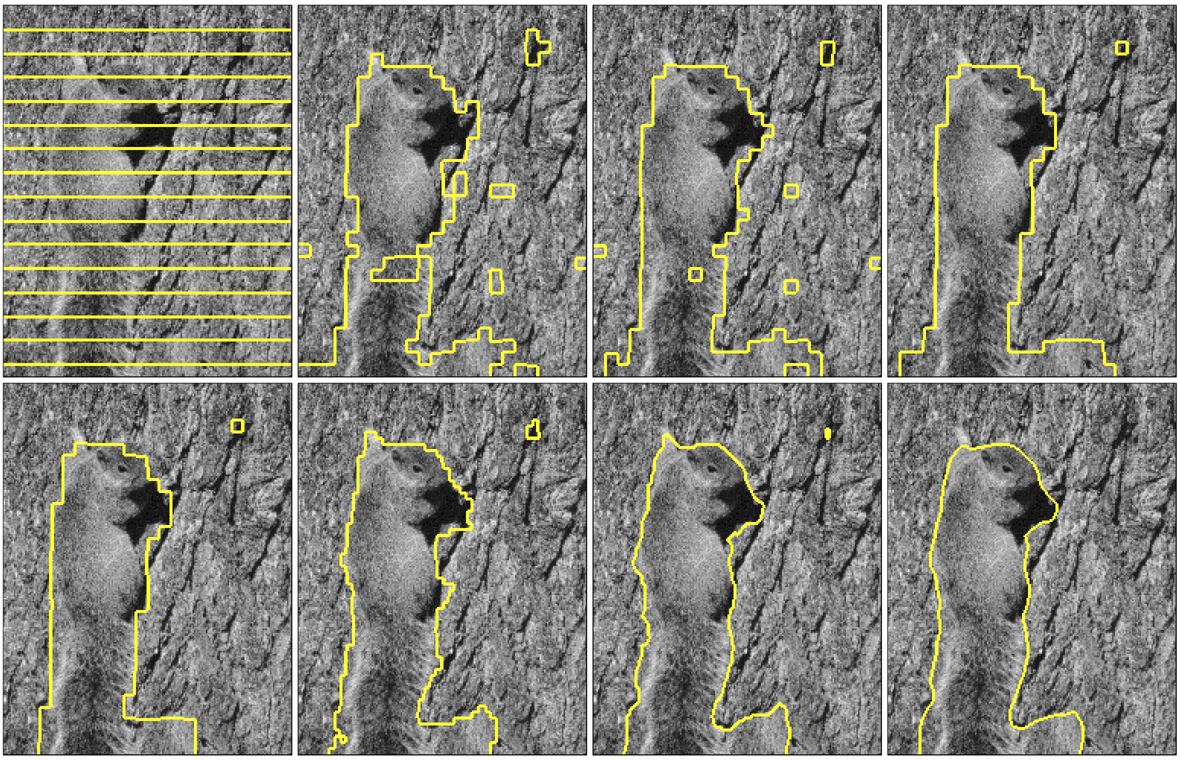


Fig. 5.45: Curve evolution including the coarse-to-fine strategy.
Top row: Scale 3 and evolution time 0, 10, 20, and 50.
Bottom row: Final result after scale 3, 2, 1, and 0.

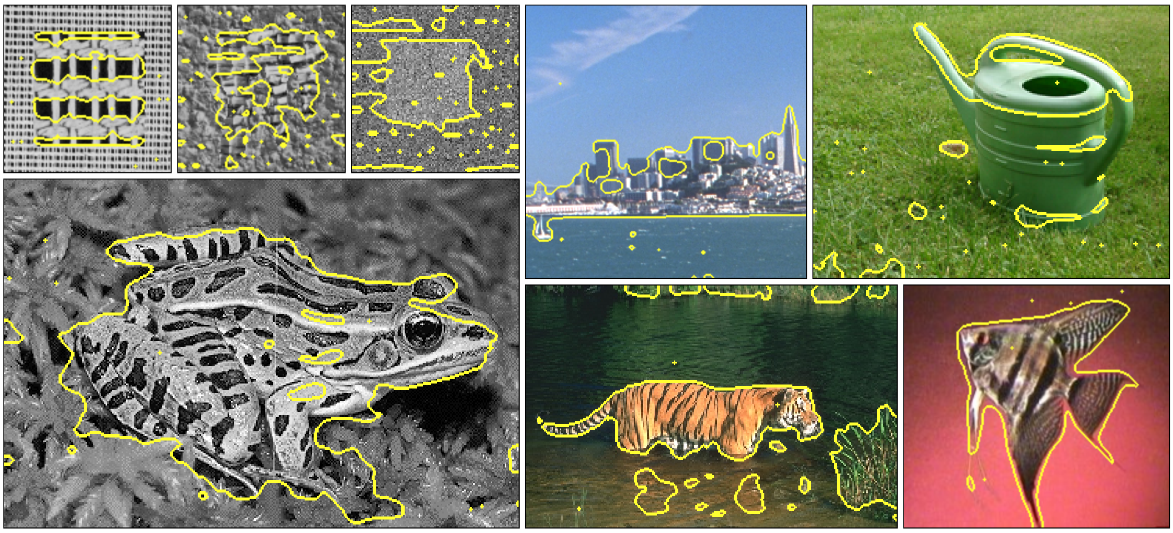


Fig. 5.46: Results without the employment of nonlinear diffusion and without a coarse-to-fine strategy. Compared to (e) in Fig. 5.16 - Fig. 5.44 some results become better, but many others get worse.

Experiments on motion segmentation. In Section 4.2 much importance has been attached to the accurate estimation of motion. It is now time to apply the derived optic flow model to motion segmentation. In contrast to the works in [CS05] and [PD05] where motion estimation and segmentation are coupled in a joint model, motion segmentation consists

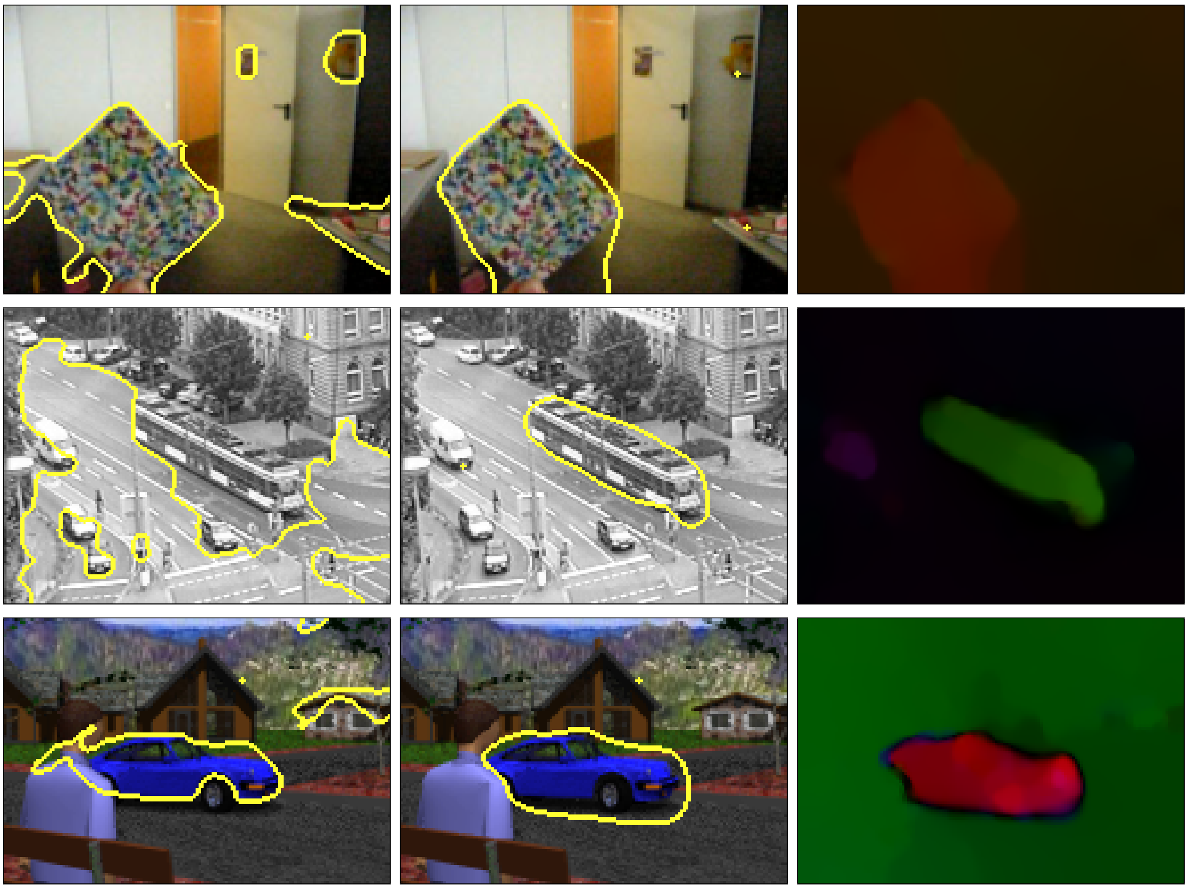


Fig. 5.47: Segmentation results for images sequences with motion. Same parameters used for all examples. LEFT: Result without motion features. CENTER: Result including motion features. BOTTOM ROW: Computed optic flow field.

here of two independent parts. One may note that this is not optimal, since there is only information flow from motion estimation to segmentation and not vice-versa. The advantage, on the other hand, is the availability of very general motion models, like the one derived in Section 4.2, whereas joint motion estimation and segmentation is so far restricted to rather simple parametric models.

Fig. 5.47 and Fig. 5.48 demonstrate the impact of the optic flow when added to the segmentation model as an additional feature. Basically the segmentation model is the same as for the texture segmentation of still images, previously described as case (a). Even the same parameters have been used.
The only novelty is the supplement of the two optic flow components to the feature vector. In order to give motion enough influence, these features have been weighted with a factor 4 in all experiments. Since there is no explanation for this specific value, it must be seen as an additional parameter. The other parameters emerging from the motion features are the parameters of the optic flow estimation method. They have been kept fixed at $\sigma = 1$, $\alpha = 50$, and $\gamma = 10$.

It can be clearly seen that motion has a very positive influence on the detection performance. In all examples, the other features are not sufficient for detecting an object in the scene. The detection by means of color and texture is disturbed by severe background clutter. The only feature that uniquely separates the object from the rest of the image is the object's motion.

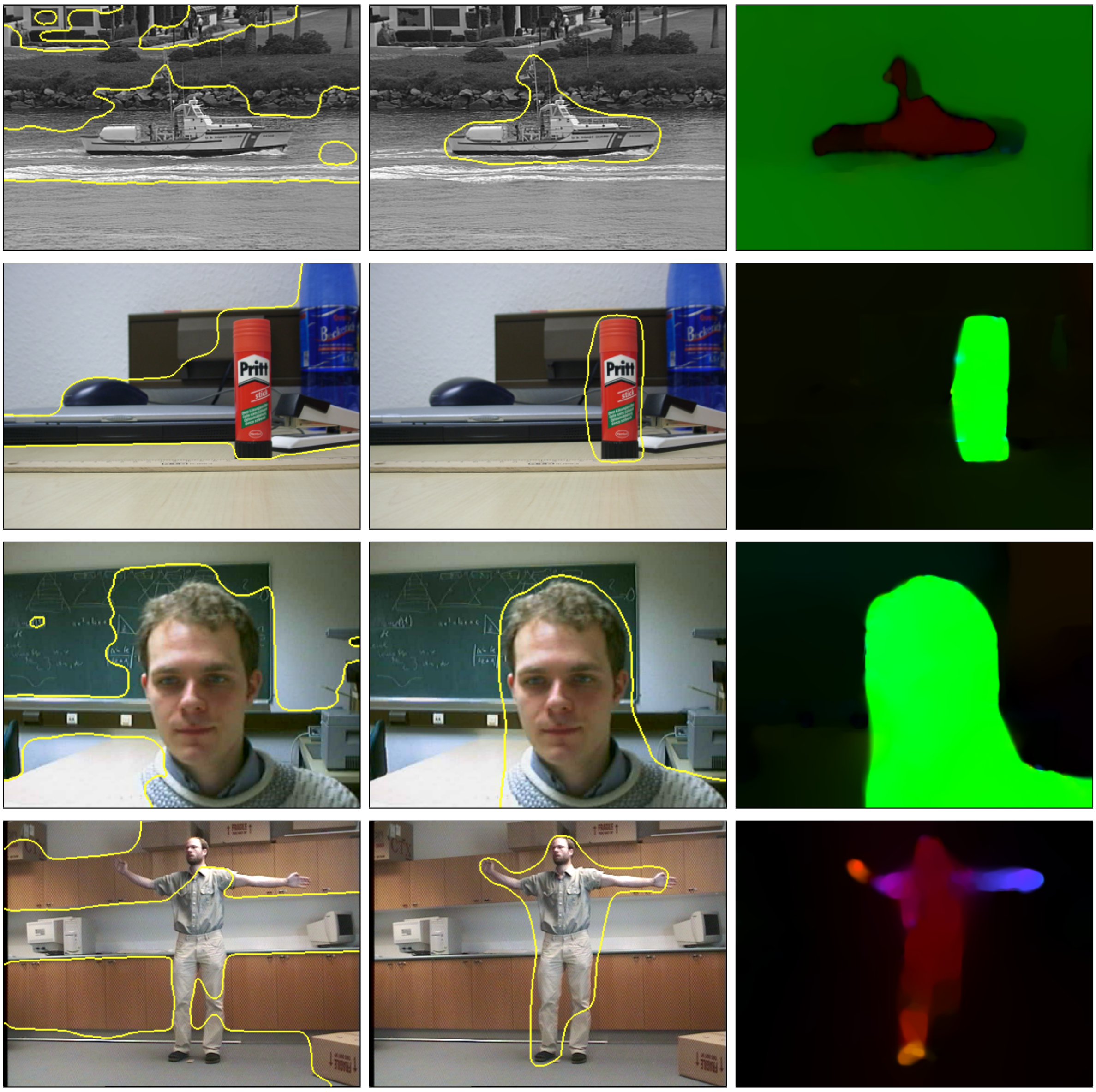


Fig. 5.48: Segmentation results for images sequences with motion. Same parameters used for all examples. LEFT: Result without motion features. CENTER: Result including motion features. BOTTOM ROW: Computed optic flow field.

However, the motion features are not only responsible for the considerably improved object detection power. There are also negative effects. In contrast to image gray value, color, or texture features, the estimated motion is far less accurate at boundaries. Even though the underlying optic flow estimation model can cope with discontinuities in the flow field, the estimated positions of these discontinuities often do not coincide with image edges. Furthermore, there are blurring artifacts in areas of occlusion or little structure, as the smoothness term of the optic flow model becomes dominant there. While these blurring artifacts do not much harm to the visual impression of the flow field, they are disastrous for the segmentation accuracy.

One can therefore draw the conclusion that optic flow features have a very positive, stabilizing effect for the coarse segmentation, but they are not well suited for an accurate boundary detection. This could lead to the following possible improvements that are left for future research:

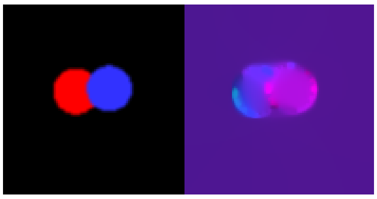


Fig. 5.49: Moving balls and flow field.

1. Reducing the weights of the motion features with finer levels in the multi-scale framework, one could exploit the good coarse detection capabilities introduced by the use of motion features while keeping the segmentation accuracy provided by the other features.
2. Furthermore, one could introduce a measure of confidence for the computed optic flow [BW05]. For the usage of color and texture features a confidence measure has not been necessary, yet with optic flow, uncertainty enters the process. Note that optic flow is actually an ill-posed problem which has only been restrained by the supplement of a smoothness term. However, this belies the fact that in some areas of the image there is simply not enough information to say anything about the motion.
   See, for instance, Fig. 5.49 in which the two balls are moving. The black background could basically move arbitrarily. Only the smoothness constraint determines the estimated flow, but it should be clear that one cannot rely on this information.
   By means of a confidence measure one allows other features in the segmentation framework to take charge in such situations without being disturbed by the unreliable motion estimates.

5.3 Level Sets and Multi-Region Segmentation

The concept of active contours embedded in the level set framework has revealed many advantages. The probably most important one is the convenient access on both the contour and the region assignment, which simultaneously allows the sound minimization of the contour length while applying advanced models for the region statistics.
Unfortunately, this image of a perfect segmentation framework is disturbed by an inherent drawback: the restriction of one embedding function to separate only two regions. Images with more than one object region are therefore no longer captured optimally by the model[6].

This problem has been addressed by several works in the past. The immediate idea to overcome the restriction is to use more than one embedding function and to assign a separate embedding function Φ_i to each region Ω_i

$$E(\Phi_i, p_i) = \int_\Omega \sum_{i=1}^{N} \left(-H(\Phi_i)\log p_i + \nu|\nabla H(\Phi_i)|\right) \, \mathbf{dx} \tag{5.42}$$

where N is the number of regions. This concept has been proposed in [MBO94, ZCMO96] and has been adopted for segmentation in [SBFAZ00, PD00a].

While the basic idea is very simple, assigning a level set function to each region entails a new difficulty that is not easy to address: the level set functions need a coupling in order to respect the constraint of disjoint regions, i.e., regions must not overlap and there must not be pixels that are not assigned to any region. Note that in the two-region case this constraint is automatically satisfied by the regions' representation.

Two different coupling concepts have been suggested. In [ZCMO96, SBFAZ00] the constraint of disjoint regions is integrated by means of a Lagrangian multiplier λ:

$$E(\Phi_i, p_i) = \int_\Omega \sum_{i=1}^{N} \left(-H(\Phi_i)\log p_i + \nu|\nabla H(\Phi_i)| + \frac{\lambda}{2}(H(\Phi_i) - 1)^2\right) \, \mathbf{dx} \tag{5.43}$$

This is a sound strategy to integrate the constraint, though it yields a rather advanced optimization problem.
Alternatively, Paragios and Deriche proposed in [PD00a] an artificial coupling force added to the evolution equation by means of an additional parameter. This methodology is easier to implement than the concept in [ZCMO96]. On the other hand it induces additional numerical parameters which can be difficult to choose.

A completely different approach has been introduced in [VC02]. Instead of assigning a separate level set function to each region, the level set functions recursively split the domain into two subdomains. This way, n level set functions can represent $N = 2^n$ regions, see Fig. 5.50. Indeed, if the number of regions is a power of 2, this multi-region model adopts from the two-region model the nice property of implicitly respecting the constraint of disjoint regions, so no further coupling forces

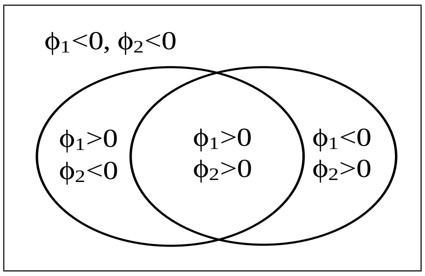


Fig. 5.50: Vese-Chan model.

[6] The ability of the level set framework to change the topology of regions should not be confused with multi-region segmentation. Although parts of a region may be disconnected, they still share the same region statistics and the same representation in the level set framework.

are necessary. The model looses parts of its attractiveness when the number of regions is not a power of 2. In such cases, parts of the region boundaries are weighted twice, and the model must introduce empty regions, which does not go well together with region models beyond the piecewise constant model.

Coupled curve evolution by means of competing regions. In the following a new coupled curve evolution will be presented. It follows the concept to use a separate level set function for each region. Instead of adding an artificial coupling force to the energy function, however, the coupling between the regions is realized in the evolution equations. Thus the task is to minimize (5.42) under the constraint $\bigcup_i \Omega_i = \Omega$ and $\bigcap_i \Omega_i = \emptyset$. Neglecting this constraint, the Euler-Lagrange equations that must be satisfied in a minimum of (5.42) read:

$$\partial_t \Phi_i = H'(\Phi_i) \left(\log p_i + \nu \operatorname{div} \left(\frac{\nabla \Phi_i}{|\nabla \Phi_i|} \right) \right). \tag{5.44}$$

Since $\log p_i$ is always negative, the curve evolution according to these equations will cause the level set functions to quickly become negative everywhere, i.e., all regions Ω_i disappear. To avoid this, it is wise to regard the concept that prevents this situation in the classical two-region segmentation.

In the two-region case where both regions are represented by a single level set function, there is always a competition between the two regions. Thus the negative value of $\log p_1$ is balanced by a positive value $\log p_2$ of the competing region. This balance is missing in (5.44). The same is true for the term resulting from the length constraint. In the two-region model, a shrinking region due to the length constraint automatically induces the competing region to capture the released area. This recapturing is missing in (5.44) as well.

With these considerations in mind, it is easy to find a way how to prevent overlap and vacuum in (5.44): one has to add a competitor to the evolution equation. If this competitor is chosen to be the most severe competitor within the narrow band there is a balanced competition between two regions, as in the classical two-region setting. The enhanced evolution equation reads

$$\partial_t \Phi_i = H'(\Phi_i) \left(\log p_i + \nu \operatorname{div} \left(\frac{\nabla \Phi_i}{|\nabla \Phi_i|} \right) - \max_{\substack{j \neq i \\ H'(\Phi_j) > 0}} \left(\log p_j + \nu \operatorname{div} \left(\frac{\nabla \Phi_j}{|\nabla \Phi_j|} \right) \right) \right). \tag{5.45}$$

One can verify easily that in the state of convergence the constraint of disjoint regions is satisfied, as each pixel is captured by one of the competing regions. This is also true for triple or quadruple junctions. In such cases each involved region competes with the best fitting region, which itself competes with the second best fitting region.

This evolution model has several advantages when compared to previous approaches:

- In contrast to the coupling by means of a Lagrange multiplier, the evolution equations stay simple and are comparatively easy to implement. As there is no update of the Lagrange multiplier necessary, the evolution can also be supposed to be more efficient.
- In contrast to the method suggested in [PD00a] the coupling does not involve any further numerical parameters that might influence the segmentation result.
- In contrast to the model in [VC02], each region, including its boundary, is directly accessible by its assigned level set function. Furthermore, the evolution is independent from the number of regions involved. There are no empty regions and no varying weights for the length constraint if the number of regions is not a power of 2.

Determining the number of regions. So far the number of regions N has been presumed to be known in advance. However, keeping the number of regions fixed from the beginning introduces a considerable degree of supervision and actually solves only a simplified version of the original segmentation problem. For a full unsupervised partitioning of the image, also the number of regions has to be determined automatically by the segmentation model.

Especially in the scope of active contours, it has usually been avoided to raise the segmentation model to this more general case. A contribution which excepts this fact is the work in [PD00a] where the number of regions is estimated in a preliminary stage by means of a Gaussian mixture estimate of the image histogram. This way, the optimum number of mixture coefficients determines the number of regions. Also the model in [VC02] can up to a certain degree deal with a variable number of regions by relying on the consideration that dispensable regions may shrink to empty regions.

An alternative way is to incorporate the sought number of regions directly into the energy functional. Interestingly it turns out that such an energy functional has already been introduced by Zhu and Yuille in [ZY96]:

$$E(\Omega_i, p_i, N) = \sum_{i=1}^{N} \left(-\int_{\Omega_i} \log p_i \, \mathbf{dx} \; + \; \frac{\nu}{2} \int_{\Gamma_i} ds \; + \; \varrho \right). \tag{5.46}$$

It extends the functionals considered so far in this chapter by introducing the number of regions N as a free variable that has to be optimized together with the shape of the regions. Furthermore, a term weighted with the parameter $\varrho > 0$ is supplemented that prevents N becoming too large.

The necessity of this additional term becomes obvious, if one regards the change of energy when a region is split into two regions. Since the region model for two separated regions can adopt much better to the underlying data than for a joint region, the energy evoked by the statistical term always decreases by a split. The energy increase due to the length constraint on the boundary is much too small to compensate for the reduced energy in the statistical term. For an acceptable balance, the weighting parameter ν has to be chosen very large, thus the evolving contour gets much too smooth. The supplement of a fixed penalty for each region allows for setting ν to reasonable values while still keeping the number of regions small.

For minimizing the energy in (5.46), a mixture of seeded region growing, region merging, and explicit active contours has been suggested in the original work. In contrast, it is sought here to minimize the energy by exploiting the advantages of the level set framework. Furthermore, the coarse-to-fine strategy coupled with a hierarchical splitting substitutes the placement of seeds.

It should be noted here that N is a discrete variable. Consequently, only minimization with respect to the contour can be performed in the classical variational framework. For optimization also with respect to the number of regions, a different strategy has to be employed.

For this optimization strategy there are two methods at hand:

- The image or a subdomain of the image can be split into two parts by the two-region segmentation framework that has been introduced in Section 5.1 and Section 5.2. Due to the usage of a coarse-to-fine strategy, the outcome of this splitting has been more or less independent from the initialization.

- By means of the evolution equation (5.45), an arbitrary set of region contours can evolve, minimizing the energy in Eq. 5.46, provided the number of regions N is fixed and reasonable initializations for the regions are available.

By combining both methods, one can efficiently minimize the energy in (5.46). Starting with the whole image domain Ω being a single region, so $N = 1$, the two-region segmentation can be applied in order to find the optimum splitting of this domain. If the energy decreases by the splitting, this results in two regions. On each of these regions, again the two-region splitting can be applied, and so on, until the energy does not decrease by further splits anymore. With this proceeding, not only the optimum number of regions is determined, but also suitable initializations for the regions.

Such a hierarchical splitting has already been proposed in combination with other segmentation techniques, e.g. the normalized cut [SM00] and semi-definite programming [KHS04]. The advantage of such a procedure is that each splitting completely ignores the cluttering rest of the image and can therefore succeed in establishing new contours without knowing good initializations.

On the other hand, ignoring parts of the image may likely not minimize the global energy. Possibilities of a region to evolve have been ignored by restricting the regions to a subdomain of the image. Contours found by early splittings determine the final result.
However, as the region number and the initialization are known, the energy can now be further optimized in the global scope by applying (5.45). This evolution adapts the regions to the new situation where they have more competitors.

The combined procedure is applied in a multi-scale setting. Starting the procedure as described on the coarsest scale, with every refinement step on the next finer scale it is checked whether any further splitting or merging decreases the energy before the evolution according to (5.45) is applied. So for each scale the optimum N as well as the region boundaries and the region statistics are updated.

In contrast to other splitting based segmentation algorithms, the method proposed here has the advantage that the contour is not fixed by an early splitting. It is allowed to evolve after the number and coarse location of regions is determined and thus all competitors are known. In contrast to other curve evolution techniques, the method is less dependent on good initializations, since the problem's complexity is temporarily reduced to two-region splits by means of the hierarchical splitting.

Experiments. In some experiments depicted in Fig. 5.51 - Fig. 5.61 the quality of this segmentation method for arbitrarily many regions has been evaluated. All parameters have been kept fixed and were the same as in Section 5.2. The additional parameter ϱ has been set to $\varrho = 0.075|\Omega|$, i.e., it has been solely adapted to the image size $|\Omega|$.

It can be seen that despite the fixed set of parameters a large variety of region numbers is detected. It reaches from 2 regions in Fig. 5.51 to 13 regions in Fig. 5.56. Furthermore, the method can deal with several real-world images that have not been partitioned satisfactorily with the method restricted to two regions.
On the other hand it is also clear that due to the increased problem complexity, some images that were segmented correctly by the two-region segmentation are over-segmented by the new method; see Fig. 5.61. A typical problem is especially the over-segmentation of smoothly varying background regions.

Two research directions are conceivable for improving the method. First the modelling accuracy of smoothly varying regions is to be improved. A more accurate description of the data in such regions may avoid unreasonable splits. Secondly, one could replace the very simple penalizer for the number of regions present in (5.46) by a more sophisticated one. In this scope one may learn from information theory and clustering techniques.

However, even these improvements will likely not ensure segmentation results that coincide with human perception in all cases. If the data provided by the image is not sufficient to distinguish regions or to keep them together, further constraints about the appearance of possible objects in the scene are absolutely necessary. This subject will be discussed in some more detail in the following section.

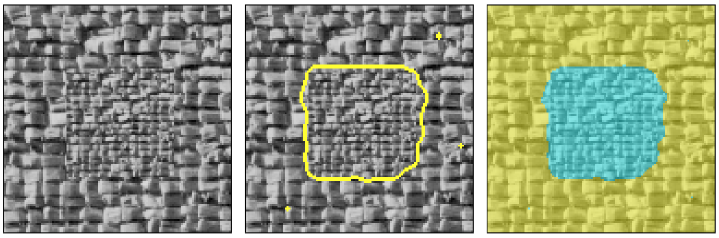


Fig. 5.51: Left: Input image (123 × 119). Center: Two-region segmentation. Right: Multi-region segmentation, 2 regions have been detected.

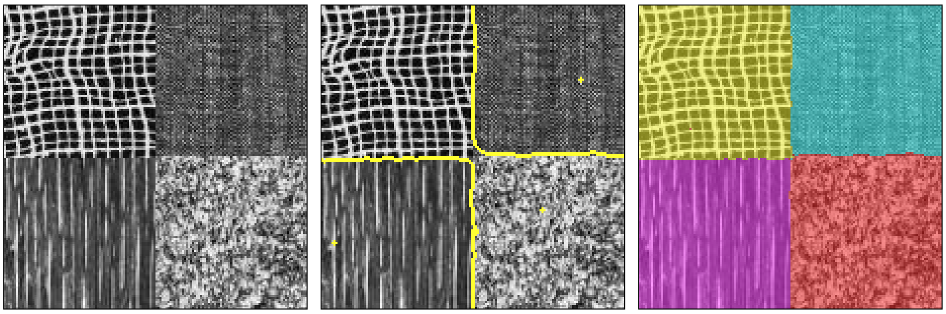


Fig. 5.52: Left: Input image (150 × 149). Center: Two-region segmentation. Right: Multi-region segmentation, 4 regions have been detected.

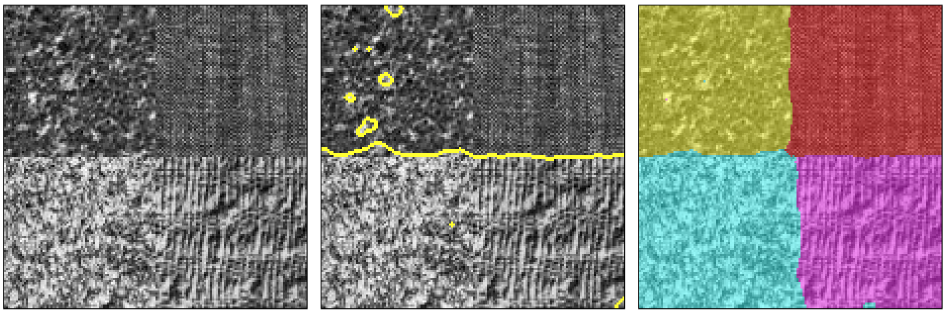


Fig. 5.53: Left: Input image (150 × 150). Center: Two-region segmentation. Right: Multi-region segmentation, 4 regions have been detected.

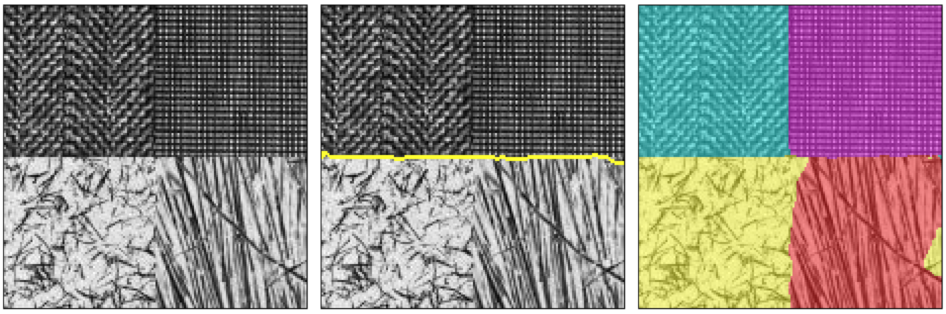


Fig. 5.54: Left: Input image (150 × 150). Center: Two-region segmentation. Right: Multi-region segmentation, 4 regions have been detected.

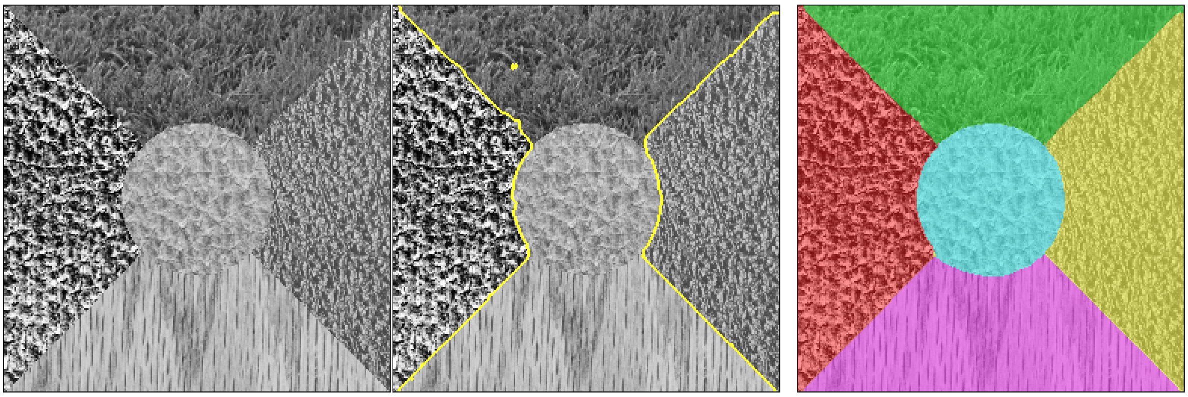


Fig. 5.55: Left: Input image (256 × 255). Center: Two-region segmentation. Right: Multi-region segmentation, 5 regions have been detected.

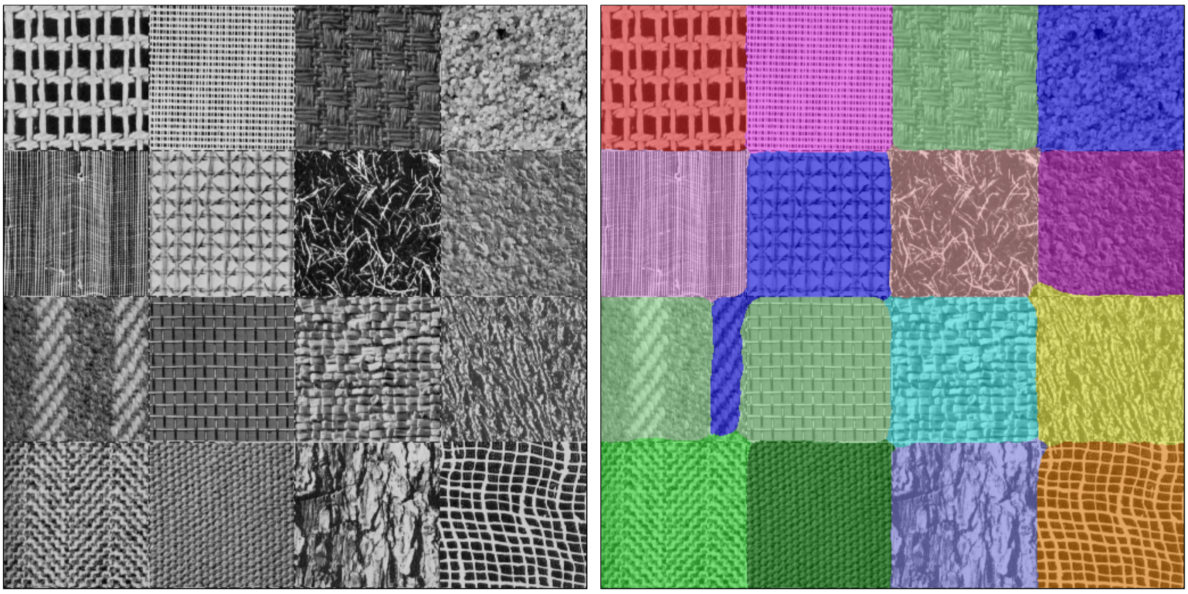


Fig. 5.56: Left: Input image (512 × 512). Right: Multi-region segmentation, 13 regions have been detected.

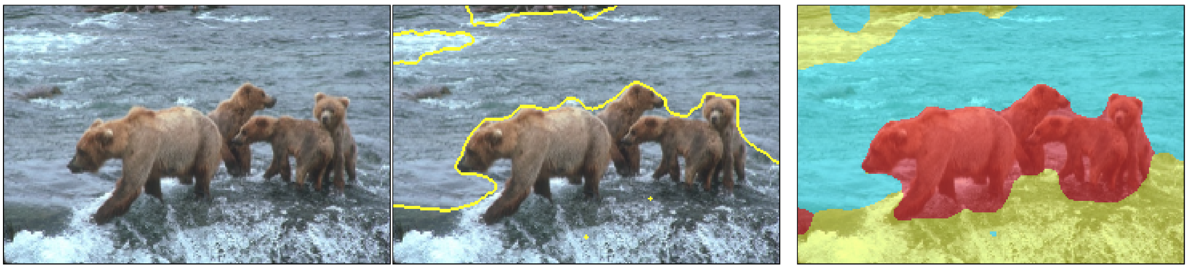


Fig. 5.57: Left: Input image (241 × 161). Center: Two-region segmentation. Right: Multi-region segmentation, 3 regions have been detected.

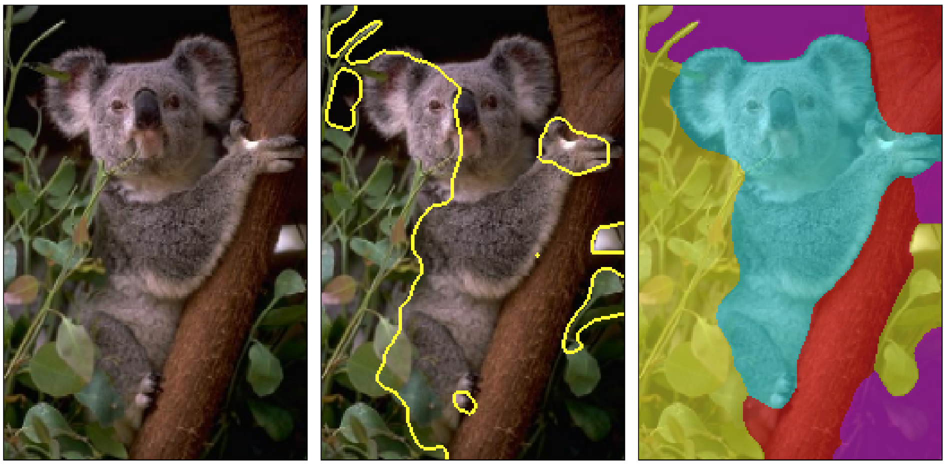


Fig. 5.58: LEFT: Input image (169 × 250). CENTER: Two-region segmentation. RIGHT: Multi-region segmentation, 4 regions have been detected.

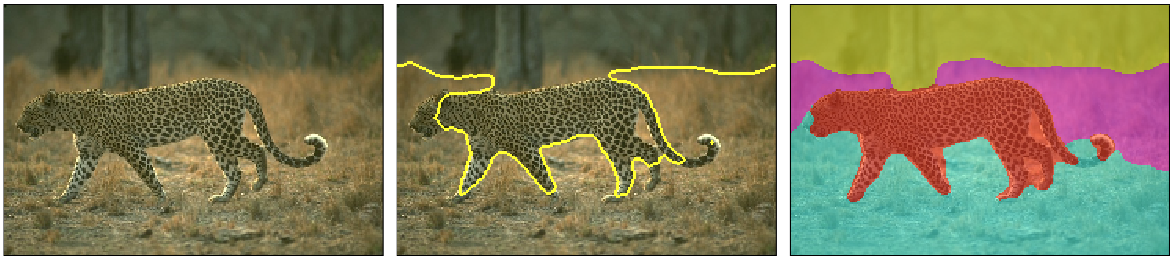


Fig. 5.59: LEFT: Input image (241 × 161). CENTER: Two-region segmentation. RIGHT: Multi-region segmentation, 4 regions have been detected.

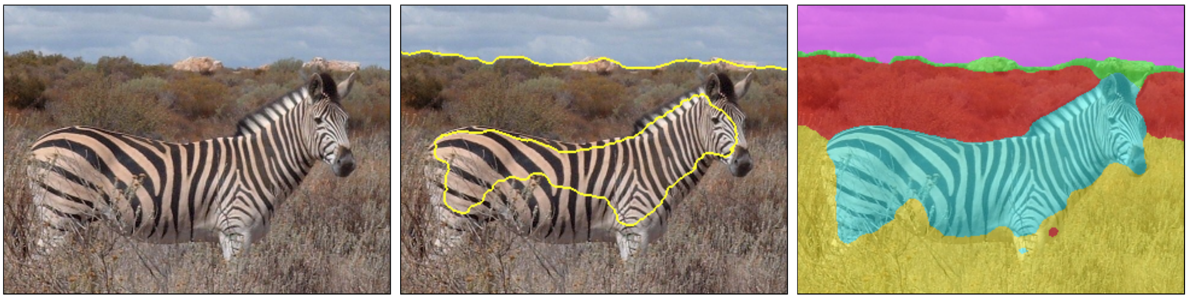


Fig. 5.60: LEFT: Input image (320 × 240). CENTER: Two-region segmentation. RIGHT: Multi-region segmentation, 4 regions have been detected.

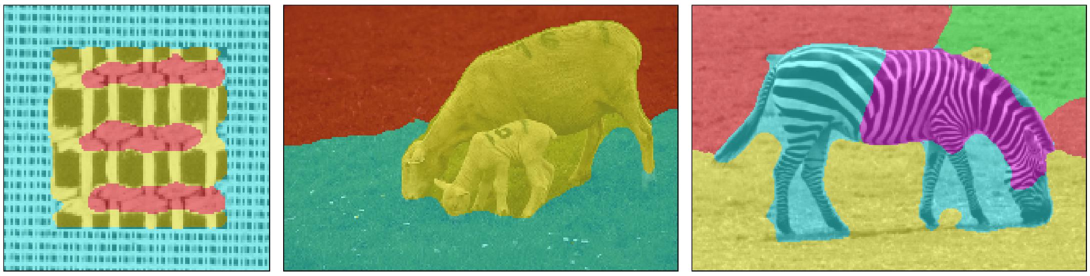


Fig. 5.61: Images that are over-segmented by the method.

5.4 3-D Shape Knowledge in Image Segmentation

The previous sections have all dealt with the objective to extract as much information as possible from the given image data in order to determine a satisfactory segmentation. Only minor, well defined constraints like smoothness assumptions have been applied in the models in order to emphasize important information and to ignore clutter and noise. In many scenes, however, the information present in the data is not sufficient for allowing these models to reliably create a good partitioning of the image. Further constraints must be supplemented in order to discard unsatisfactory solutions.

Whereas such top-down constraints are often implicitly present in many algorithms that are specifically designed for a certain task, this final section aims on the *explicit* integration of a top-down constraint into the segmentation model. Explicit constraints integrated by means of a quantitative measure have two important advantages over implicit algorithmic constraints:

- The model assumption which is responsible for the top-down constraint is clearly stated, i.e., it is evident under which circumstances the model works satisfactorily and when it must be supposed to fail. Unknown side effects of the algorithm in certain situations are avoided.
- An explicit top-down constraint usually keeps the model more general. In the ideal case, the model can be adapted to new situations without changing the algorithm itself, but solely by changing some model data. In principle, this may allow to learn certain prior knowledge automatically in the future.

A very well suited constraint for image segmentation can be provided by the introduction of shape knowledge. This is, one searches object contours that comply with the shape of a certain object or a certain set of objects. For the segmentation model this means that it additionally has to determine the object including its pose that fits best to the data given in the image. In return, the shape knowledge may discard many unsatisfactory solutions.

The introduction of shape knowledge in active contour based image segmentation goes back on the work in [LGF00] where the curve evolution has been biased by a term that draws the curve to a given shape:

$$\partial_t \Phi = g\, |\nabla \Phi| \operatorname{div}\left(\frac{\nabla \Phi}{|\nabla \Phi|}\right) + \nabla g^\top \nabla \Phi + \lambda(\Phi_0 - \Phi). \qquad (5.47)$$

Apart from the new bias term, this is exactly equation (5.20) for the evolution of geodesic active contours. The new term, weighted by a parameter $\lambda > 0$, directs the curve towards the prior shape Φ_0 given as a level set function. Moreover, the shape Φ_0 is translated and rotated at each evolution step in order to best fit the current curve estimate Φ.

This basic concept has been extended and modified in various proximate works [CSW01, CTT$^+$01, RP02, Cre02, CTT$^+$02, CTWS02, CKS03, PRR03, CSS04, COS04, RRKS04]. The main subjects of these works are how to introduce shape knowledge in a variational framework, how to estimate shape transformation parameters (in particular [RP02, PRR03, RRKS04, COS04]), how to model statistical variations of the shape knowledge, and how to apply shape knowledge in the presence of multiple objects with different shapes [CSS04].

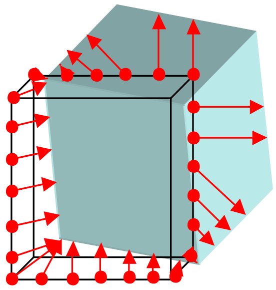
All these works have in common that they apply 2-D shape models. This appears natural, as also image segmentation in general works on the scene projected to the 2-D image plane. An extension to 3-D segmentation in the presence of volumetric data using the level set framework is straightforward [RPD04].

Contrary to these works, the concept here is the integration of a 3-D object shape into a 2-D segmentation model. The motivation for this idea is immediate: albeit images are 2-D structures, they show projected objects from the real world, which is three-dimensional. While a 2-D shape model does not take into account that different views of the object produce different projected shapes in the image, a 3-D shape model automatically contains the projections of all possible object views. Moreover, the pose estimation, which fits the shape model to the image data, not only allows for the integration of shape knowledge into segmentation, but additionally yields the position of the object in 3-D space including its distance from the camera. The only prerequisite is a calibrated camera[7].

The expected advantageous property of a model with integrated shape knowledge is its capability to deal with cluttered scenes. On the other hand, the model contains more unknowns and so the associated optimization problem becomes much more difficult to solve, particularly with respect to local optima. Although the global optimum might be a very satisfactory solution, it can be very difficult to efficiently find this optimum. The research described here in this section is still at an early stage. For this reason, the model is so far restricted to extract exactly one specified object from the scene. Furthermore, a rough pose initialization of the object is given at the beginning.

3-D pose estimation by means of a 2-D contour. The most interesting issue of the approach is the 3-D pose estimation and how it can be coupled with the presented segmentation model. The pose estimation method is completely adopted from Bodo Rosenhahn. Those of his works that are relevant for this section can be found in [Ros03, RS04, RKS04, RPS05]. In the following, a very brief description of the underlying ideas will be given before the pose estimation method will be coupled with the segmentation model.

Goal of the pose estimation is to translate and rotate the 3-D object model, which is given by a set of 3-D mesh points of the object surface, such that its projection to the image plane best fits the segmented object region in the image. Provided a perfect segmentation and optimized pose parameters, the projection of the object surface to the image plane yields exactly the same region as the segmentation. If the segmentation is defective or if the object in the image is not exactly the same as in the object model, however, the regions do not match perfectly anymore. There appears a non-overlapping error which is sought to be minimized.



Given the projection matrix of the camera, a region in the image determined by some segmentation process, and an initialization of the 6 pose parameters[8], one can search for correspondences between points on the object surface and points in the image region. Since only correspondences between points on the contour of the image region and points on the contour of the region provided by the projected object surface contain useful information, possible correspondences are restricted to be established between these subsets of points.
Correspondences are then determined by assigning each point on the region contour to the next available surface point. Thereby, there are two possibilities to measure the distance between these 2-D and 3-D entities. Either one projects the surface points to the image plane

[7] A related work is stereoscopic segmentation [YS01, GM04] where images from multiple calibrated cameras are used for a 3-D segmentation. The resulting 3-D shape yields also the depth of each surface point and therefore a 3-D reconstruction of the segmented object.

[8] 3 degrees of freedom for the rotation and 3 for the translation

and measures the distances in 2-D, or one transforms the 2-D contour points to lines in 3-D and measures the closest distance of surface points to these lines. The method of Rosenhahn relies on the latter 3-D distance measure.

Note that no matter which way the distance between corresponding points is computed, it involves a transformation between 2-D and 3-D entities. Rosenhahn therefore notes the pose estimation method in the language of Clifford algebras. The line reconstructed from a contour point is expressed as a Plücker line $\mathbf{L} = (\mathbf{n}, \mathbf{m})$ where $\mathbf{n}$ is a unit direction and $\mathbf{m}$ the moment of the line. The rigid motion, i.e., the translation and rotation of a surface point $\mathbf{X}$ in homogeneous 4-D coordinates is expressed as matrix $M = \exp(\theta\xi)$ where $\theta\xi$ contains the pose parameters. The line $\mathbf{L}$ reconstructed from the contour point can then be compared to the transformed surface point $M\mathbf{X}$ by considering the perpendicular error vector from $M\mathbf{X}$ to $\mathbf{L}$:

$$\mathbf{d} = (M\mathbf{X})_3 \times \mathbf{n} - \mathbf{m} \tag{5.48}$$

where $(\cdot)_3$ takes the first three components of a homogeneous 4-D vector and × denotes the cross product.

Each correspondence between a contour point and a surface point thus yields by (5.48) three equations which are nonlinear due to M. Rosenhahn proposed to decompose the nonlinear equations into a recursive sequence of linearized equations. Applying a least squares approach to the linearized equations leads to an over-determined linear system which can be solved, e.g., by the Householder method [PTVF92]. Iteration finally yields the optimized pose parameters for the nonlinear equations.

In [RKS04] Rosenhahn extended this object model by including kinematic chains. The pose estimation method can hence not only deal with rigid objects but also with objects that have additional degrees of freedom at well-defined joints.
Instead of a single global screw motion, parts of the object are allowed to perform a different motion than the main body, provided they respect the joint constraints. The motion of a kinematic chain consisting of n joints is described by a set of transformation matrices $M_1(\theta_1\xi_1), ..., M_n(\theta_n\xi_n)$. For a surface point of the outer part of such a kinematic chain, the error vector from (5.48) then takes the form

$$\mathbf{d} = (M_n \dots M_1 M\mathbf{X})_3 \times \mathbf{n} - \mathbf{m}. \tag{5.49}$$

The joint constraints are coded in the matrices ξ_i. Only the scalars θ_i are free parameters that yield additional unknowns for the pose estimation task and appear in the above-mentioned linear system. Note that only correspondences of surface points from an outer part of the kinematic chain yield constraints for estimating the additional degrees of freedom.
For further details about the outlined pose estimation method the reader is referred to [Ros03] and [RPS05].

Joining pose estimation and segmentation. Obviously, the described pose estimation technique relies on contour points in the image that have to be extracted by some segmentation method. On the other hand, segmentation approaches need the pose estimation in order to utilize shape knowledge. This interdependence between the two vision tasks clearly suggests a joint approach where both the pose parameters and the object contour in the image are optimized at once.

Starting point is the segmentation functional (5.39), which describes the two-region segmentation model with the possibility to use color, texture, and motion cues. In order to utilize the shape knowledge, a shape term is added to this functional. It penalizes deviations of the contour from the expectation given by the projected shape model:

$$E(\Phi,\theta\xi) = \int_\Omega \left(-\sum_{j=1}^{M} \Big(H(\Phi)\log p_{1,j} - (1-H(\Phi))\log p_{2,j} \Big) + \nu|\nabla H(\Phi)| + \underbrace{\lambda\left(\Phi - \Phi_0(\theta\xi)\right)^2}_{\text{shape}} \right) \mathbf{dx}. \quad (5.50)$$

Hereby it is assumed that the projected shape is given as contour by means of the embedding function Φ_0. The weighting parameter $\lambda \geq 0$ allows to adjust the influence of the shape knowledge on the segmentation outcome.

Obviously, the energy of this new functional not only depends on the contour, but as well on the pose parameters of the shape model. This is because the distance $(\Phi - \Phi_0)^2$ can be minimized both by adapting Φ and by changing the pose, thus changing Φ_0. While changes in Φ_0 are constrained to correspond to screw motions of the object model, changes in Φ are constrained by the image data and the contour length.

The shape penalizer is motivated from the Bayes formula already used to derive the segmentation model without shape knowledge (see Section 5.1):

$$P(\mathbf{x}\in\Omega_i|I(\mathbf{x})=s) = \frac{P(I(\mathbf{x})=s|\mathbf{x}\in\Omega_i)P(\mathbf{x}\in\Omega_i)}{P(I(\mathbf{x})=s)}.$$

In Section 5.1, the unconditional probability $P(\mathbf{x}\in\Omega_i)$ for the pixel to belong to region Ω_i has only depended on the length of the boundary. Now with a shape model available, $P(\mathbf{x}\in\Omega_i)$ can be modelled in a much more sophisticated way. Assuming the shapes are Gaussian distributed with the given object shape Φ_0 as mean and constant standard deviation $\sigma = \frac{1}{\sqrt{\lambda}}$,

$$p(\Phi(\mathbf{x})) \propto \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(\Phi(\mathbf{x})-\Phi_0(\mathbf{x}))^2}{2\sigma^2}\right) \quad (5.51)$$

maximization of $P(\mathbf{x}\in\Omega_i|I(\mathbf{x})=s)$, and equivalently minimization of its negative logarithm, leads to (5.50)[9].

It is obviously easy to introduce more sophisticated statistical models of the shape prior. Instead of the simple Gaussian distribution with a fixed standard deviation, one could as well employ a Gaussian with variable standard deviation [RP02] or a nonparametric kernel estimate [COS04].
Note the important role of the level set representation as it transcribes a spatial distance between contours to a tonal distance by means of a distance transform. Also note that the probabilistic modelling allows the segmentation to disengage from the object model.

Joint optimization. For optimizing (5.50), the pose parameters are to be adapted to the most recent contour Φ. The opposite way, the contour has to evolve taking into account the image data and the projected shape model with the most recent pose parameters. This yields an iterative approach that alternates optimization of the pose parameters and curve evolution. Both iteration steps thereby seek to minimize the last term of (5.50).

Pose estimation given a contour has been described above. After a set of correspondences between contour points and surface points has been determined, a gradient descent solves for

[9] One could certainly argue that the shape prior obsoletes the length constraint on the contour, yet for the time being it is still kept in the functional for the segmentation to yield smooth results.

the unknown pose parameters. Also the evolution of Φ is performed as usual by a gradient descent. The evolution equation now contains an additional force acting on the curve:

$$\partial_t \Phi = H'(\Phi) \left(\sum_{j=1}^{M} (\log p_{1,j} - \log p_{2,j}) + \nu \operatorname{div} \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) \right) + \underbrace{\lambda (\Phi_0 - \Phi)}_{\text{shape}} . \tag{5.52}$$

Since the object model is not given as a 2-D shape but as a 3-D surface model, it has to be projected onto the image plane to yield Φ_0. Let X_S denote the set of points $\mathbf{X}$ on the surface of the object. Projection of X_S onto the image plane yields the set x_S of all homogeneously scaled 2-D points $\mathbf{x}$ on the image plane that correspond to a 3-D point in the surface model

$$\mathbf{x} = P\,\mathbf{X}, \qquad \forall \mathbf{X} \in X_S \tag{5.53}$$

where P denotes the projection matrix of the camera. Furthermore, the surface points may undergo a screw motion $M = \exp(\theta\xi)$ with the pose parameters $\theta\xi$. This leads to

$$\mathbf{x} = P\,M\,\mathbf{X}, \qquad \forall \mathbf{X} \in X_S. \tag{5.54}$$

The level set function Φ_0 can be constructed from x_S by setting

$$\Phi_0(\mathbf{x}) = \begin{cases} 1 & : \text{ if } \mathbf{x} \in x_S \\ -1 & : \text{ else} \end{cases} \tag{5.55}$$

and applying a distance transform. Note that the distance transform ensures that the penalty in the energy functional corresponds to the error measure used in the pose estimation algorithm.

Experiments. The joint segmentation and pose estimation is tested in two different scenarios: in the first one, the pose of a tea box is estimated in two scenes with considerable background clutter. In the second scenario, the object model is extended to kinematic chains and the pose of the human upper body is estimated again in a scene with inhomogeneous background and significant illumination effects.

The segmentation model of the previous sections relied mainly on global region statistics, since they can be computed more efficiently than local statistics. Usage of shape knowledge changes this situation, as it becomes possible to deal with a cluttered background. However, global statistics do not provide much useful information anymore, if the regions are inhomogeneous. Local statistics, on the other hand, only assume homogeneity of a region within a local window and can represent cluttered background regions, as well as an inhomogeneous object region, much more accurately. For this reason, the global statistics are consistently substituted by local statistics as presented in Section 5.1.2.

The other segmentation parameters have been taken from the previous sections. Only the parameter ν for the length constraint of the boundary has been considerably reduced by factor 5. This is due to the fact that its task is now mainly accomplished by the more sophisticated shape constraint. The new parameter λ for weighting the shape constraint has been set to $\lambda = \frac{1}{3}$ for the tea box sequences and $\lambda = 1$ for integrating the human body model.

Fig. 5.62 demonstrates the joint segmentation and pose estimation. The initialization is fairly far away from the true pose of the tea box. However, the segmentation adapts the contour to the data in the image and thereby also forces the pose of the object to be adapted. It can be

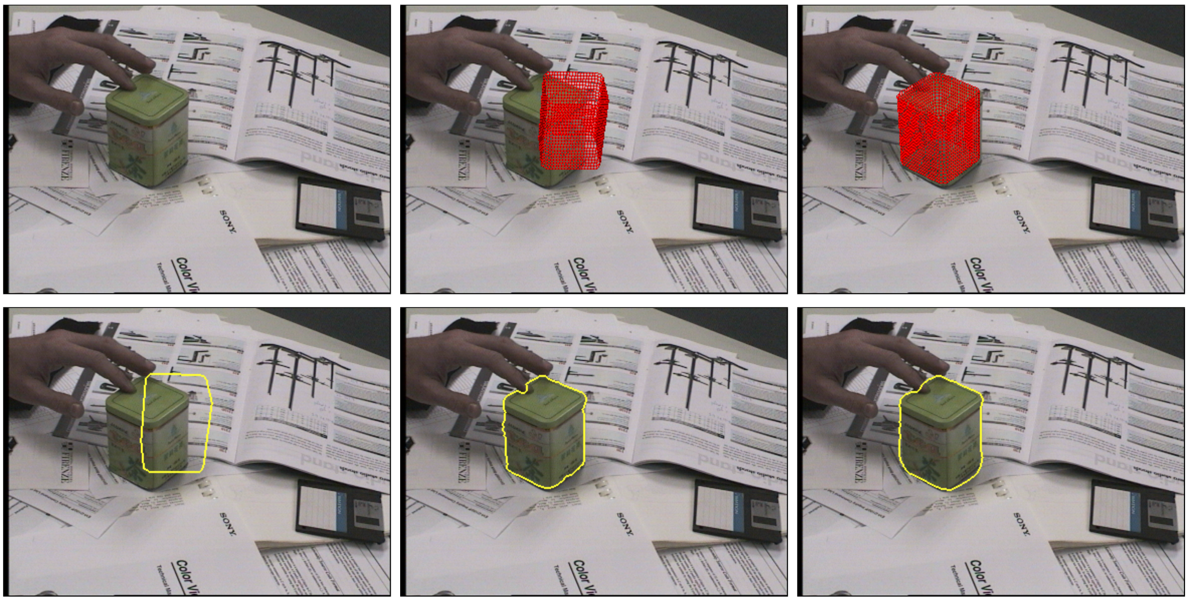


Fig. 5.62: Top Row: Tea box (384 × 288). Initialization of object pose. Estimated pose. Bottom Row: Region initialization. Segmentation result. Segmentation without shape knowledge.

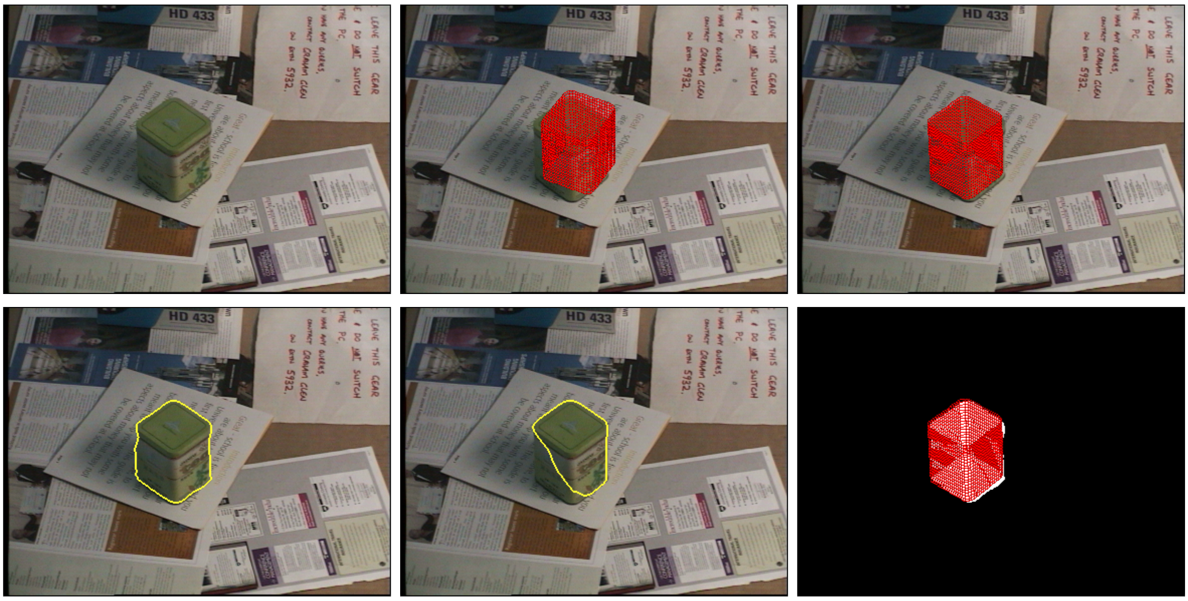


Fig. 5.63: Top Row: Tea box (384 × 288). Initialization of object pose. Estimated pose. Bottom Row: Segmentation result. Segmentation without shape knowledge. Segmentation and mesh.

seen that in this scene the shape knowledge is not necessary to find the correct segmentation. The initialization in the vicinity of the object is in this case already sufficient.

This situation changes in Fig. 5.63. Despite the close initialization, the segmentation is no longer able to capture the complete tea box without the help of the object model. It is also shown how well the projected object surface matches the segmented region. However, one can also see that the region is not completely identical to the projected shape. This is due to the probabilistic modelling of the shape by means of (5.51) and a non-zero standard deviation[10]. It allows the segmentation to deviate from the shape model if necessary.

[10] In fact, the standard deviation is determined by the inverse of the parameter λ.

This flexibility, however, is at the same time a shortcoming of the current approach. A demonstration is depicted in Fig. 5.64. To the same extend as the region is allowed to grow beyond the projected surface, the segmentation can also produce errors. If the pose estimation follows these errors, the region can again grow a bit further, which finally causes the object to drift away. On the other hand it is also not possible to set the standard deviation in (5.51) to 0. This would cause λ to be infinite, i.e., the image data has no influence anymore and both segmentation and pose estimation can no longer adapt to the image data. A reasonable solution to this problem might be the usage of robust error norms, which would possibly keep the object model from following the erroneous segmentation. This is subject to further research.

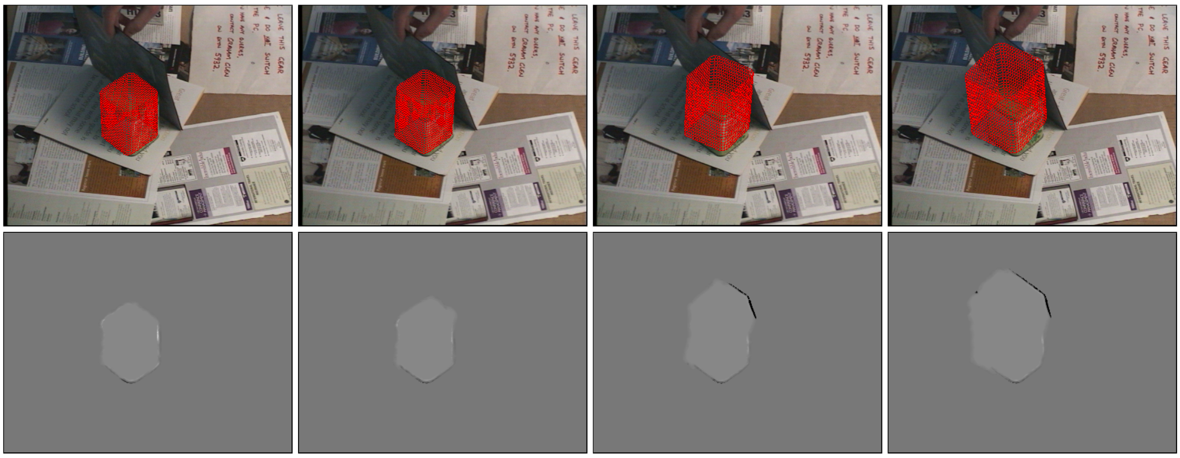


Fig. 5.64: TOP ROW: Pose estimation for images 62 - 65 of the tea box sequence. BOTTOM ROW: Corresponding level set functions. As soon as the segmentation produces larger errors, pose estimation fails, and the object prior is no longer a reliable anchor that can constrain the segmentation.

Fig. 5.65 finally demonstrates the capabilities of the technique in the presence of a more complex object model containing kinematic chains. Human pose estimation is a very popular research topic [GD96, BM98, FPT01, ST03]. While the human body is often modelled by means of ellipsoids, the approach presented here employs free form surfaces that allow a much more accurate modelling. Thus the model fits much more precisely to the observations in the image.

Apart from the object model containing kinematic chains, the approach is further extended by employing two cameras calibrated to the same coordinate system. Segmentation in both images yields a further contour which provides additional correspondences between contour points and surface points. After pose estimation, the object model is projected to both image planes in order to constrain the segmentations there. The concept can easily be extended to scenarios with an arbitrary number of cameras.

Again, despite the close initialization, segmentation without shape knowledge fails. Segmentation constrained by the human model, on the other side, extracts a rather good contour bearing the considerable background clutter and shading effects in mind. Only the right hand of the person is not captured correctly due to the poor contrast between the arm and the closet. This also influences the pose estimation in the first frame. During the further tracking of the body, however, also the right hand is captured correctly.

This last example shows the performance and the generality of the approach. Although there are still many open problems concerning the robustness of joint segmentation and pose estimation illustrated in Fig. 5.64, the technique is a good basis for further interesting research. In particular the incorporation of geometric issues establishes connections between so far hardly related research areas.

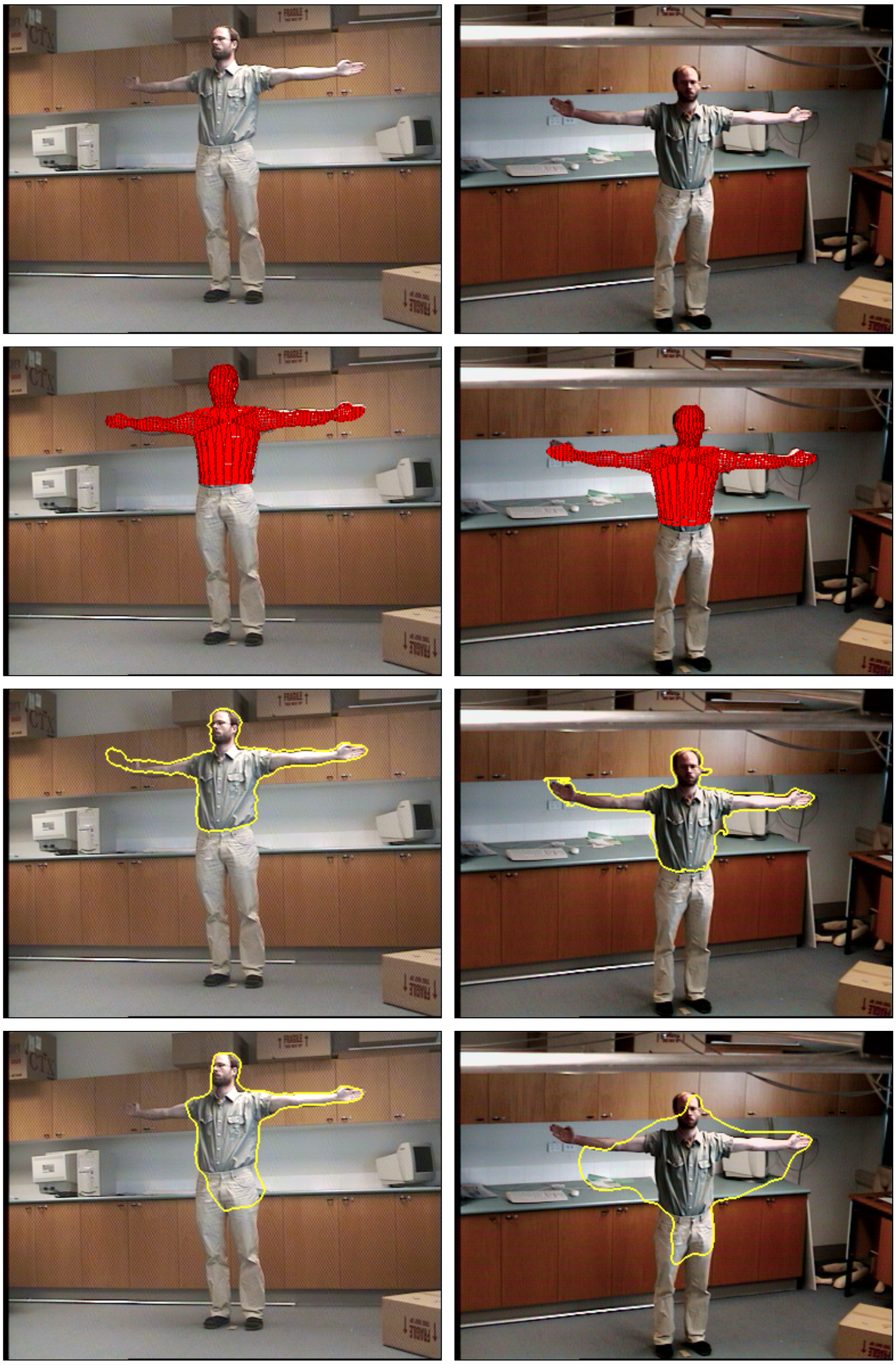
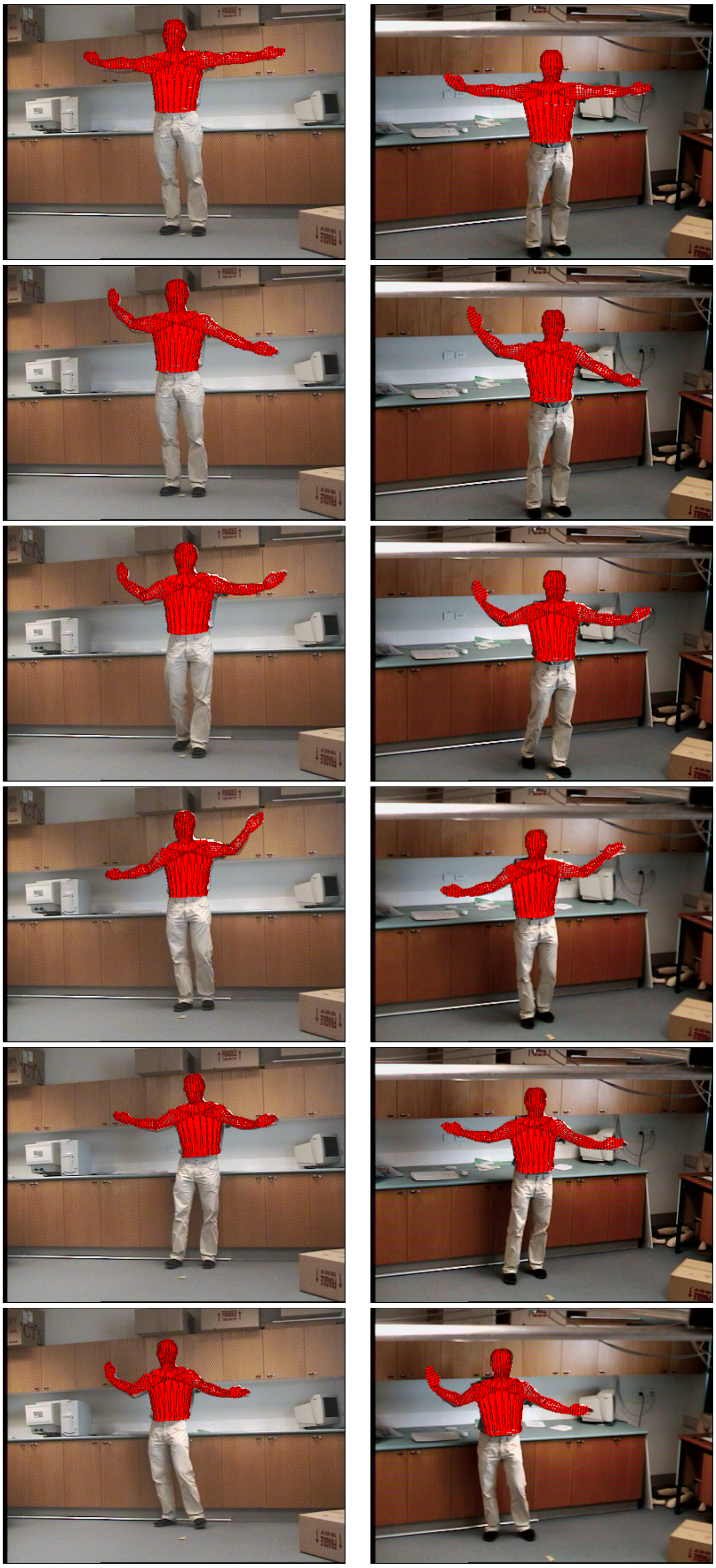


Fig. 5.65: Segmentation and pose estimation of the human upper body using two cameras.
FIRST ROW: First stereo image of the sequence (384 × 288).
SECOND ROW: Initialization of the pose.
THIRD ROW: Segmentation result. FOURTH ROW: Segmentation without shape knowledge.
NEXT PAGE: Estimated pose at images 1, 11, 21, 31, 41, and 51 of the sequence.



6 Summary and Outlook

Opportunity is missed by most people because
it comes dressed in overalls and looks like work.

Thomas Edison (1847-1931)
Inventor

This thesis has dealt with many contemporary applications of partial differential equations in image analysis. Chapter 2 started with PDEs in image denoising, both by means of nonlinear diffusion filters and variational methods. Interestingly it turned out that the strong relations between both denoising methods even yield an equivalence in the special case of space-discrete, one-dimensional TV flow and TV regularization. Altogether, TV flow and TV regularization have proven to be outstanding denoising techniques with nice properties. They have been successfully applied for improving orientation estimation by means of the structure tensor and for optic flow estimation within a variational framework. Moreover, TV flow has lead to an interesting new region based scale measure. This wide spread of applications gives TV flow an important position in image analysis, even comparable to the position of Gaussian convolution.

Another application of TV flow has been investigated in Chapter 3 in the scope of numerical approximations of hyperbolic conservation laws. TV flow has been employed there for removing oscillation artifacts of the classical second order Lax-Wendroff method. However, the strategy to use stabilized inverse diffusion filters for improving the accuracy of the first order Upwind scheme has been much more successful. The derived technique produces results with a similar accuracy as modern TVD methods.

Based upon the findings on TV flow and its application in the nonlinear structure tensor, Chapter 4 looked into the tasks of low level vision aside from image denoising. In the scope of texture analysis it has been demonstrated that the structure tensor for estimating orientation combined with a local measure for scale provides a texture feature space that can act as a sparse alternative to the popular Gabor feature space.

In the scope of optic flow estimation TV flow has been applied both in a local method and in form of a TV regularizer in a global variational method. However, the research on optic flow estimation presented in Chapter 4 goes far beyond the application of TV minimization ideas. Two fundamental problems in differential optic flow estimation have been addressed here: firstly the problem of illumination changes in the image sequence; this problem has been addressed by introducing a constancy assumption that is largely invariant under illumination changes, such as the image gradient; secondly the problem of large displacements, which has been addressed by no longer linearizing the constancy assumptions. The proposed numerical scheme for minimizing the energy functional emerging from this optic flow model not only deals with local optima in the functional by introducing a continuation method, it also establishes a relation between energy functionals with non-linearized constancy assumptions and so-called warping methods. It thereby provides a theoretic foundation for warping methods, which have so far only been algorithmically motivated. Apart from this theoretic contribution, the method also yields optic flow estimates that are more accurate than all results presented in the literature so far.

Finally in Chapter 5, image segmentation in the sense of contour extraction has been studied as a further application of PDE methods. A segmentation model based on active contours in the level set framework has been combined with various statistical models for describing the interior of regions. Furthermore, additional cues besides the image gray value have been incorporated. In detail these are the texture and motion cues derived in Chapter 4. The statistical modelling in a Bayesian framework thereby ensures that the more discriminative features automatically gain more weight than features that provide poor discriminative information in a certain image. In this scope it has been shown that the usage of a full Gaussian probability density model ensures the discrimination properties of a feature to be contrast independent, which is in sharp contrast to the widely used Gaussian model with a fixed standard deviation as known from the Mumford-Shah functional.

Since image segmentation in general causes like optic flow estimation severe multi-modal optimization problems, again a continuation method has been proposed in order to encourage convergence to the global optimum. Experiments on a larger set of images while using strictly the same parameters showed that this strategy works quite reliably.

Subsequently, a new coupled evolution equation has been presented that allows the joint evolution of more than one level set function respecting the requirement of non-overlapping regions and the need for every pixel to be assigned to a region. With this framework it is also possible to extend the classical two-region segmentation with active contours to multiple-region segmentation. Thereby the number of regions is determined by a hierarchical strategy which also provides rather reliable initializations for the regions.

This purely data-driven segmentation has been finally extended by introducing a top-down shape constraint to the segmentation model. In contrast to other segmentation approaches with incorporated shape knowledge, the shape model is given as a three-dimensional object surface and is projected onto the image plane in order to constrain the segmentation there. This involves a 3-D pose estimation problem which is solved jointly with the segmentation by means of a pose estimation algorithm developed by Rosenhahn.

These techniques dealing with a wide spread of image analysis tasks show the importance of partial differential equations in contemporary image processing and computer vision. This is not too surprising, as PDE based methods provide both a sound mathematical foundation *and* very good results. Although this conclusion is not new, it may always be worth a reminder.

Future Research. This thesis may have given many insights into image analysis and has presented several new ideas. However, a general rule in science says that answering one question just rises two others. This is also the case here. In this final paragraph I would like to draw the attention on four research directions that, starting from this work, appear most promising to me.

In Chapter 4 a lot of effort has been put into the accurate computation of optic flow. However, the utilization of this motion information for segmentation purposes has so far only been negligently been implemented by just using the flow vector as a feature like color and texture; yet optic flow is different from these features. In contrast to color and texture, optic flow is only an estimate, i.e., it can contain errors which are completely different from typical noise

models. In some areas of the image optic flow may be more reliable than in others. This fact should be reflected in a more sophisticated integration model.
Furthermore, the current integration model focuses mainly on discontinuities in the flow field, though discontinuities are the most inaccurate part of optic flow. Putting more emphasis on utilizing the absolute motion vector may be much more successful.

Aside the more sophisticated integration of motion information, there is a lot of potential to improve the multi-region segmentation method presented in Section 5.3. Higher order region models may avoid slowly varying regions to become split unnecessarily. The non-trivial task is thereby the combination of such models with the presented statistical representation of the regions' interior. Besides this, other ways to determine the number of regions may prove to be superior. Suitable concepts may hereby be adopted from information theory. Alternatively, prior object knowledge could decide whether a region should be split or not.

Generally, the utilization of object knowledge offers the opportunity for many new ideas. In particular the presence of 3-D concepts in the suggested joint segmentation and pose estimation approach from Section 5.4 creates connections to techniques like 3-D reconstruction from stereo images, structure from motion, shape from shading, and shape from texture. They all generate depth information that can improve pose estimation. The other way round, the object model might in a top-down manner provide important cues for these techniques. Especially the tracking of an object once it has been detected can be expected to become much more robust this way.

The other branch of research in the presence of object knowledge is the autonomous detection of objects in a scene. The contour obtained by means of segmentation is only one feature that describes an object, and in cluttered scenes, it had to be initialized appropriately to find the object in the image. There are certainly many other object features, yet the important questions are, which of these are reliable in a certain situation, how they can be represented, and how they can be combined in order to emphasize always the most useful feature. This general object recognition task might be one of the most challenging topics of image analysis in the forthcoming years.

A Equivalence of TV Flow and TV Regularization

In this chapter, the proofs for the analytic solutions of TV flow and TV regularization in the space-discrete 1-D case are provided. Since it turns out that both analytic solutions are identical, it is proven at the same time that space-discrete TV flow and TV regularization are equivalent in 1-D. The proofs in this chapter have been worked out in collaboration with Martin Welk and Gabriele Steidl.

A.1 Proof: Analytical Solution for Space-Discrete TV Flow

Consider a space-discrete formulation of TV diffusion. Further assume the spatial grid size to be 1 and let $f = (f_0, \ldots, f_{N-1})$ denote a discrete version of the input signal $f(x)$ with N pixels. This leads to the following dynamical system:

$$\left.\begin{aligned}\dot{u}_0 &= \operatorname{sgn}(u_1 - u_0), \\ \dot{u}_i &= \operatorname{sgn}(u_{i+1} - u_i) - \operatorname{sgn}(u_i - u_{i-1}) \qquad (i = 1, \ldots, N-2), \\ \dot{u}_{N-1} &= -\operatorname{sgn}(u_{N-1} - u_{N-2}), \\ u(0) &= f.\end{aligned}\right\} \tag{A.1}$$

In the following, $u_{-1} := u_0$ and $u_N := u_{N-1}$, which may be regarded as a discretization of the homogeneous Neumann boundary conditions. Since the right-hand side of this system is discontinuous, a more detailed specification of when a system of functions is said to satisfy these differential equations (cf. also [Fil88]) is needed. A vector-valued function u is said to fulfill the system in (A.1) over the time interval $[0, T]$ if the following holds true:

(I) u is an absolutely continuous vector-valued function which satisfies (A.1) almost everywhere, where sgn is defined by $\operatorname{sgn} w := 1$ if $w > 0$, $\operatorname{sgn} w := -1$ if $w < 0$, and may take any value in $[-1, 1]$ if $w = 0$.

(II) If $\dot{u}_i(t)$ and $\dot{u}_{i+1}(t)$ exist for the same t, and $u_{i+1}(t) = u_i(t)$ holds, then the expression $\operatorname{sgn}(u_{i+1}(t) - u_i(t))$ occurring in both the right-hand sides for $\dot{u}_i(t)$ and $\dot{u}_{i+1}(t)$ must take the same value in both equations.

Under these conditions one obtains the following result:

Proposition 1 (Properties of Space-Discrete TV Diffusion)
The system in (A.1) *has a unique solution $u(t)$ in the sense of (I) and (II). This solution has the following properties:*

(i) (Finite Extinction Time)
There exists a finite time $T \geq 0$ such that for all $t \geq T$ the signal becomes constant:

$$u_i(t) = \frac{1}{N} \sum_{k=0}^{N-1} f_k \qquad \textit{for all } i = 0, \ldots, N-1.$$

(ii) (Finite Number of Merging Events)
There exists a finite sequence $0 = t_0 < t_1 < \ldots < t_{n-1} < t_n = T$ *such that the interval* $[0, T)$ *splits into sub-intervals* $[t_j, t_{j+1})$ *with the property that for all* $i = 0, \ldots, N-2$ *either* $u_i(t) = u_{i+1}(t)$ *or* $u_i(t) \neq u_{i+1}(t)$ *throughout* $[t_j, t_{j+1})$*. The absolute difference between neighboring pixels does not become larger for increasing* $t \in [t_j, t_{j+1})$*.*

(iii) (Analytical Solution)
In each of the sub-intervals $[t_j, t_{j+1})$ *constant regions of* $u(t)$ *evolve linearly:*
For a fixed index i *consider a constant region given by*

$$u_{i-l+1} = \ldots = u_i = u_{i+1} = \ldots = u_{i+r} \qquad (l \geq 1, r \geq 0) \tag{A.2}$$

and

$$u_{i-l} \neq u_{i-l+1} \text{ if } i-l \geq 0, \quad u_{i+r} \neq u_{i+r+1} \text{ if } i+r \leq N-1$$

for all $t \in [t_j, t_{j+1})$*. We call* (A.2) *a region of size* $m_{i,t_j} = l + r$*. For* $t \in [t_j, t_{j+1})$ *let* $\triangle t = t - t_j$*. Then* $u_i(t)$ *is given by*

$$u_i(t) = u_i(t_j) + \mu_{i,t_j} \frac{2\triangle t}{m_{i,t_j}} \, ,$$

where μ_{i,t_j} *reflects the relation between the region containing* u_i *and its neighboring regions. It is given as follows:*
For inner regions (i.e. $i - l \geq 0$ *and* $i + r \leq N - 1$*) we have*

$$\mu_{i,t_j} = \begin{cases} 0 & \text{if} \quad (u_{i-l}, u_i, u_{i+r+1}) \textit{ is strictly monotonous}, \\ 1 & \text{if} \quad u_i \textit{ is minimal in } (u_{i-l}, u_i, u_{i+r+1}), \\ -1 & \text{if} \quad u_i \textit{ is maximal in } (u_{i-l}, u_i, u_{i+r+1}) \end{cases} \tag{A.3}$$

and in the boundary case ($i - l + 1 = 0$ *or* $i + r = N - 1$)*, the evolution is half as fast:*

$$\mu_{i,t_j} = \begin{cases} 0 & \text{if} \quad m = N, \\ \frac{1}{2} & \text{if} \quad u_i \textit{ is minimal in } (u_{i-l}, u_i, u_{i+r+1}), \\ -\frac{1}{2} & \text{if} \quad u_i \textit{ is maximal in } (u_{i-l}, u_i, u_{i+r+1}). \end{cases} \tag{A.4}$$

Proof.
Let u be a solution of (A.1). It is shown that u is uniquely determined and satisfies the rules (i)–(iii). The proof proceeds in four steps.

1. If $\dot{u}(t)$ exists at a fixed time t and $u_i(t)$ lies at this time in some region

$$u_{i-l+1}(t) = \ldots = u_i(t) = \ldots = u_{i+r}(t) \qquad (l \geq 1, r \geq 0),$$

$$u_{i-l}(t) \neq u_{i-l+1}(t) \text{ if } i-l \geq 0, \quad u_{i+r}(t) \neq u_{i+r+1}(t) \text{ if } i+r \leq N-1$$

of size $m_{i,t}$, then it follows by (A.1) and (II) in the non-boundary case $i - l \geq 0$ and $i + r \leq N - 1$ that

$$u_i(t) \;=\; \frac{1}{m_{i,t}} \sum_{k=-l+1}^{r} u_{i+k}(t),$$

and therefore

$$\begin{aligned}\dot{u}_i(t) &= \frac{1}{m_{i,t}} \sum_{k=-l+1}^{r} \dot{u}_{i+k}(t) \\ &= \frac{1}{m_{i,t}} \left(\operatorname{sgn}\left(u_{i+r+1}(t) - u_i(t)\right) \;-\; \operatorname{sgn}\left(u_i(t) - u_{i-l}(t)\right)\right) \\ &= \mu_{i,t} \, \frac{2}{m_{i,t}}\,, \end{aligned} \tag{A.5}$$

where $\mu_{i,t}$ describes the relation between the region containing u_i and its neighbors at time t as in (A.3). In the boundary case $i - l + 1 = 0$ or $i + r = N - 1$ one can follow the same lines and obtains (A.5) with $\mu_{i,t}$ defined by (A.4).

2. Let $\dot{u}(t)$ exist in some small interval (τ_0, τ_1) and assume that $u_i(t) \neq u_{i+1}(t)$ for some $i \in \{0, \ldots, N-2\}$ and all $t \in (\tau_0, \tau_1)$. By continuity of u one may assume that $u_i(t) < u_{i+1}(t)$ throughout (τ_0, τ_1). The opposite case $u_i(t) > u_{i+1}(t)$ can be handled the same way. Then we obtain by (A.5) and definition of $\mu_{i,t}$ for all $t \in (\tau_0, \tau_1)$ that

$$\dot{u}_i(t) \;\geq\; 0 \quad \text{if} \quad i - l \geq 0, \tag{A.6}$$

$$\dot{u}_i(t) \;>\; 0 \quad \text{if} \quad i - l + 1 = 0, \tag{A.7}$$

$$\dot{u}_{i+1}(t) \;\leq\; 0 \quad \text{if} \quad i + r \leq N - 2, \tag{A.8}$$

$$\dot{u}_{i+1}(t) \;<\; 0 \quad \text{if} \quad i + r = N - 1. \tag{A.9}$$

Set $w(t) := u_{i+1}(t) - u_i(t)$. Then the mean value theorem yields

$$w(\tau_1) - w(\tau_0) = (\tau_1 - \tau_0)\, \dot{w}(t^*)$$

for some $t^* \in (\tau_0, \tau_1)$ and one gets by (A.6)–A.9 that

$$w(\tau_1) - w(\tau_0) \leq 0$$

with strict inequality in the boundary case. Consequently, the difference between pixels cannot become larger in the considered interval. In particular, by continuity of u, pixels cannot be split. Once merged they stay merged.

3. Now we start at time $t_0 = 0$. Let t_1 be the largest time such that $\dot{u}(t)$ exists and no merging of regions appears in $(0, t_1)$. Then, for all $i \in \{0, \ldots, N-1\}$, a function u_i is in the same region with the same relations to its neighboring regions throughout $[0, t_1)$. Thus, we conclude by (A.5) that

$$\dot{u}_i(t) = \mu_{i,0} \, \frac{2}{m_{i,0}} \qquad (t \in (0, t_1))$$

and consequently

$$\begin{aligned} u_i(t) &= \mu_{i,0} \, \frac{2t}{m_{i,0}} + C_{i,0} \\ &= f_i \;+\; \mu_{i,0} \, \frac{2t}{m_{i,0}} \qquad (t \in [0, t_1]), \end{aligned}$$

where the last equality follows by continuity of u_i if t approaches 0.

4. We are now in the position to analyze the entire chain of merging events successively. Next we consider the largest interval (t_1, t_2) without merging events in the same way, taking the initial setting $u(t_1)$ instead of f into account. Then we obtain
$$u_i(t) = \mu_{i,t_1} \frac{2t}{m_{i,t_1}} + C_{i,t_1},$$
where $u_i(t_1) = \mu_{i,t_1} \frac{2t_1}{m_{i,t_1}} + C_{i,t_1}$ by continuity of u_i. Consequently
$$u_i(t) = u_i(t_1) + \mu_{i,t_1} \frac{2(t-t_1)}{m_{i,t_1}}.$$
One can continue in the same way by considering $[t_2, t_3)$ and so on. Since there is only a finite number N of pixels and some of these pixels merge at the points t_j the process stops after a finite number of n steps with output
$$u_i(t_n) = \frac{1}{N} \sum_{k=0}^{N-1} f_k$$
for all $i = 0, \ldots, N-1$.

Conversely, it is easy to check that a function u with (i)–(iii) is a solution of the system in (A.1). This completes the proof of the proposition. □

A.2 Proof: Analytical Solution for Discrete TV Regularization

Next it is proven that discrete TV regularization satisfies the same rules as space-discrete TV diffusion. For given initial data $f = (f_0, \ldots, f_{N-1})$ discrete TV regularization consists in constructing the minimizer
$$u(\alpha) = \min_u E(u; \alpha, f) \tag{A.10}$$
of the functional
$$E(u; \alpha, f) = \sum_{i=0}^{N-1} \left((u_i - f_i)^2 + 2\alpha \left| u_{i+1} - u_i \right| \right), \tag{A.11}$$
where again Neumann boundary conditions are supposed, thus $u_{-1} = u_0$ and $u_N = u_{N-1}$. For a fixed regularization parameter $\alpha \geq 0$, the minimizer of (A.11) is uniquely determined since $E(u; \alpha, f)$ is strictly convex in $u_0, \ldots, u_{N-1}$. Furthermore, $E(u, \alpha; f)$ is a continuous function in $u_0, \ldots, u_{N-1}, \alpha$. Consequently, $u(\alpha)$ is a (componentwise) continuous function in α.

The following proposition implies together with Proposition 1 the equivalence of space-discrete TV diffusion and discrete TV regularization, if the diffusion time t is identical to the regularization parameter α.

Proposition 2 (Properties of Discrete TV Regularization)
The function $u(\alpha)$ in (A.10) *is uniquely determined by the following rules:*

(i) (Finite Extinction Parameter)
There exists a finite $A \geq 0$ such that for all $\alpha \geq A$ the signal becomes constant:
$$u_i(\alpha) = \frac{1}{N} \sum_{k=0}^{N-1} f_k \qquad \textit{for all } i = 0, \ldots, N-1.$$

(ii) (Finite Number of Merging Events)
There exists a finite sequence $0 = a_0 < a_1 < \ldots < a_{n-1} < a_n = A$ *such that the interval* $[0, A)$ *splits into sub-intervals* $[a_j, a_{j+1})$ *with the property that for all* $i = 0, \ldots, N-2$ *either* $u_i(\alpha) = u_{i+1}(\alpha)$ *or* $u_i(\alpha) \neq u_{i+1}(\alpha)$ *throughout* $[a_j, a_{j+1})$*. The absolute difference between neighboring pixels does not become larger for increasing* $\alpha \in [a_j, a_{j+1})$*.*

(iii) (Analytical Solution)
In each of the sub-intervals $[a_j, a_{j+1})$ *constant regions of* $u(\alpha)$ *evolve linearly:*
For a fixed index i *let us consider a constant region given by*

$$u_{i-l+1} = \ldots = u_i = u_{i+1} = \ldots = u_{i+r} \qquad (l \geq 1, r \geq 0) \tag{A.12}$$

and

$$u_{i-l} \neq u_{i-l+1} \text{ if } i-l \geq 0, \quad u_{i+r} \neq u_{i+r+1} \text{ if } i+r \leq N-2 \tag{A.13}$$

for all $\alpha \in [a_j, a_{j+1})$*. We call* (A.12) *a region of size* $m_{i,a_j} = l + r$*. For* $\alpha \in [a_j, a_{j+1})$ *let* $\triangle\alpha = \alpha - a_j$*.*
Then $u_i(\alpha)$ *is given by*

$$u_i(\alpha) = u_i(a_j) + \mu_{i,a_j} \frac{2\triangle\alpha}{m_{i,a_j}},$$

where μ_{i,a_j} *reflects the relation between the region containing* u_i *and its neighboring regions. It is given as follows:*
For inner regions (i.e. $i - l \geq 0$ *and* $i + r \leq N - 2$*) we have*

$$\mu_{i,a_j} = \begin{cases} 0 & \text{if} \quad (u_{i-l}, u_i, u_{i+r+1}) \textit{ is strictly monotonous,} \\ 1 & \text{if} \quad u_i \textit{ is minimal in } (u_{i-l}, u_i, u_{i+r+1}), \\ -1 & \text{if} \quad u_i \textit{ is maximal in } (u_{i-l}, u_i, u_{i+r+1}) \end{cases} \tag{A.14}$$

and in the boundary case ($i - l + 1 = 0$ *or* $i + r = N - 1$*), the evolution is half as fast:*

$$\mu_{i,a_j} = \begin{cases} 0 & \text{if} \quad m = N, \\ \frac{1}{2} & \text{if} \quad u_i \textit{ is minimal in } (u_{i-l}, u_i, u_{i+r+1}), \\ -\frac{1}{2} & \text{if} \quad u_i \textit{ is maximal in } (u_{i-l}, u_i, u_{i+r+1}). \end{cases} \tag{A.15}$$

Proof:
Again our proof proceeds in four steps. It has a similar structure as the proof of Proposition 1.

1. Let us first verify the solution $u(\alpha)$ of (A.10) for an arbitrary but fixed $\alpha > 0$.
   If $u_i(\alpha)$ is contained in some region of size $m_{i,\alpha}$ with (A.12), A.13, then, in case $i - l \geq 0$ and $i + r \leq N - 2$, $u(\alpha)$ can be obtained as minimizer of

$$\begin{aligned} & E(u_0, \ldots, u_{i-l}, u_i, u_{i+r+1}, \ldots, u_{N-1}; \alpha, f) \\ &= \sum_{k=-l+1}^{r} (u_i - f_{i+k})^2 + 2\alpha \left(|u_i - u_{i-l}| + |u_{i+r+1} - u_i| \right) \\ &\quad + F(u_0, \ldots, u_{i-l}, u_{i+r+1}, \ldots, u_{N-1}) \end{aligned}$$

with some function F independent of u_i. By (A.12), A.13 the partial derivative of E with respect to u_i exists and is given by

$$\frac{\partial E}{\partial u_i} = 2 \sum_{k=-l+1}^{r} (u_i - f_{i+k}) \; - \; 4\alpha\mu_{i,\alpha}.$$

Here $\mu_{i,\alpha}$ describes the relation between the region containing u_i and its neighbors for the regularization parameter α as in (A.14). Setting the partial derivative to zero, we obtain

$$u_i(\alpha) = \frac{1}{m_{i,\alpha}} \sum_{k=-l+1}^{r} f_{i+k} \; + \; \mu_{i,\alpha} \frac{2\alpha}{m_{i,\alpha}} \, . \tag{A.16}$$

In the boundary case $i - l + 1 = 0$ or $i + r = N - 1$ one can follow the same lines and obtains (A.16) with $\mu_{i,\alpha}$ defined by (A.15).

2. Next it is shown that initially merged pixels will not be split for any α in a small interval $[0, a_1]$.

   For $\alpha = 0$ it holds $u(0) = f$. Let $f_i = u_i(0)$ be contained in some region of the form

   $$f_{i-l_0+1} = \ldots = f_i = f_{i+1} = \ldots = f_{i+r_0} \qquad (l_0, r_0 \geq 1)$$

   and

   $$f_{i-l_0} \neq f_{i-l_0+1} \text{ if } i - l_0 \geq 0, \quad f_{i+r_0} \neq f_{i+r_0+1} \text{ if } i + r_0 \leq N - 2.$$

   By continuity of $u(\alpha)$ we can choose $\alpha_1 > 0$ so that $u_i(\alpha) \neq u_{i-l_0}(\alpha)$ and $u_{i+1}(\alpha) \neq u_{i+r_0}(\alpha)$ throughout $[0, \alpha_1)$. Assume that there exists $\alpha \in (0, \alpha_1)$ so that $u_i(\alpha) \neq u_{i+1}(\alpha)$, where it may be assumed that

   $$u_i(\alpha) < u_{i+1}(\alpha). \tag{A.17}$$

   The opposite case $u_i(\alpha) > u_{i+1}(\alpha)$ can be handled the same way. Note that at time α more pixels than u_i and u_{i+1} may be separated. However, by (A.16) with some $1 \leq l \leq l_0$ and some $1 \leq r \leq r_0$ it holds that

   $$\begin{aligned} u_i(\alpha) &= \frac{1}{l} \sum_{k=-l+1}^{0} f_{i+k} \; + \; \mu_{i,\alpha} \frac{2\alpha}{l} \; = \; f_i \; + \; \mu_{i,\alpha} \frac{2\alpha}{l} \, , \\ u_{i+1}(\alpha) &= \frac{1}{r} \sum_{k=1}^{r} f_{i+k} \; + \; \mu_{i+1,\alpha} \frac{2\alpha}{r} \; = \; f_i \; + \; \mu_{i+1,\alpha} \frac{2\alpha}{r} \, , \end{aligned}$$

   where we see by (A.17) and (A.14), A.15 that $\mu_{i,\alpha} \geq 0$ and $\mu_{i+1,\alpha} \leq 0$. Thus, $u_i(\alpha) \geq u_{i+1}(\alpha)$ which contradicts (A.17). Consequently $u_i(\alpha) = u_{i+1}(\alpha)$ throughout $[0, \alpha_1)$, i.e., the pixels of the initial region stay merged.

   Let $a_1 > 0$ denote the largest number such that no merging of regions appears in $[0, a_1)$. Then we have for all $i = 0, \ldots, N - 1$ and all $\alpha \in [0, a_1)$ that $\mu_{i,\alpha} = \mu_{i,0}$ and regarding that $u(\alpha)$ is continuous that

   $$u_i(\alpha) = f_i \; + \; \mu_{i,0} \frac{2\alpha}{m_{i,0}} \qquad (\alpha \in [0, a_1]). \tag{A.18}$$

3. Now it is shown that the absolute difference between neighboring regions cannot become larger with increasing $\alpha \in [0, a_1)$.
Without loss of generality let for some fixed index i

$$u_{i-l+1} = \ldots = u_i < u_{i+1} = \ldots = u_{i+r} \qquad (l, r \geq 1)$$

and

$$u_{i-l} \neq u_{i-l+1} \text{ if } i - l \geq 0, \quad u_{i+r} \neq u_{i+r+1} \text{ if } i + r \leq N - 2.$$

Consider the non-boundary case $i - l \geq 0$ and $i + r \leq N - 2$ first. By (A.18) one obtains for $\alpha + \delta \in [0, a_1)$, $\delta > 0$ that

$$\begin{array}{rcl} d_i(\alpha) &=& u_{i+1}(\alpha) - u_i(\alpha) \;=\; f_{i+1} - f_i + 2\alpha \left(\frac{\mu_{i+1,0}}{r} - \frac{\mu_{i,0}}{l} \right), \\ d_i(\alpha + \delta) &=& u_{i+1}(\alpha + \delta) - u_i(\alpha + \delta) \;=\; f_{i+1} - f_i + 2(\alpha + \delta) \left(\frac{\mu_{i+1,0}}{r} - \frac{\mu_{i,0}}{l} \right) \end{array}$$

and consequently

$$d_i(\alpha + \delta) - d_i(\alpha) = 2\delta \left(\frac{\mu_{i+1,0}}{r} - \frac{\mu_{i,0}}{l} \right).$$

By (A.14) it follows that

$$\frac{\mu_{i+1,0}}{r} - \frac{\mu_{i,0}}{l} = \begin{cases} 0 & \text{if } u_{i-l} < u_i < u_{i+1} < u_{i+r+1}, \\ -\frac{1}{r} & \text{if } u_{i-l} < u_i \text{ and } u_{i+1} > u_{i+r+1}, \\ -\frac{1}{l} & \text{if } u_{i-l} > u_i \text{ and } u_{i+1} < u_{i+r+1}, \\ -\frac{1}{r} - \frac{1}{l} & \text{if } u_{i-l} > u_i \text{ and } u_{i+1} > u_{i+r+1} \end{cases}$$

which yields the desired property $d_i(\alpha) \geq d_i(\alpha + \delta)$.

In case of boundary regions we follow the same lines but replace (A.14) by (A.15). Then one can see that the absolute difference between neighboring regions becomes smaller with increasing $\alpha \in [0, a_1)$.

4. We are now in the position to analyze the entire chain of merging events successively. For $\alpha > a_1$ and $\triangle\alpha = \alpha - a_1$, consider

$$\tilde{u}_i(\triangle\alpha) = \min_u E(u; \triangle\alpha, u(a_1)).$$

The same considerations as in Part 2 of the proof can be repeated, but with initial setting $u(a_1)$ instead of f. It follows that there exists a_2 such that for all $i = 0, \ldots, N-2$ either $\tilde{u}_i(\triangle\alpha) = \tilde{u}_{i+1}(\triangle\alpha)$ or $\tilde{u}_i(\triangle\alpha) \neq \tilde{u}_{i+1}(\triangle\alpha)$ throughout $[a_1, a_2)$, where the absolute difference between neighboring pixels does not become larger for increasing $\triangle\alpha$. Further, we obtain by (A.18) and (A.16) that

$$\begin{array}{rcl} \tilde{u}_i(\triangle\alpha) &=& u_i(a_1) + \mu_{i,a_1} \dfrac{2\triangle\alpha}{m_{i,a_1}} \\ &=& \dfrac{1}{m_{i,a_1}} \displaystyle\sum_{j \in R_{i,a_1}} f_j + \mu_{i,a_1} \dfrac{2a_1}{m_{i,a_1}} + \mu_{i,a_1} \dfrac{2\triangle\alpha}{m_{i,a_1}}, \end{array}$$

where $R_{i,\alpha} = \{j : u_j(\alpha) \text{ is in the region of } u_i(\alpha)\}$ while m_{i,a_1} denotes the size of the region containing $u_i(a_1)$ and μ_{i,a_1} reflects the relation between the region containing

$u_i(a_1)$ and its neighboring regions. Since the relations between regions do not change for $\triangle\alpha \in [0, a_2 - a_1)$ we can rewrite $\tilde{u}_i(\triangle\alpha)$ as

$$\begin{aligned}\tilde{u}_i(\triangle\alpha) &= \frac{1}{m_{i,a_1+\triangle\alpha}} \sum_{j\in R_{i,a_1+\triangle\alpha}} f_j + \mu_{i,a_1+\triangle\alpha}\frac{2(a_1+\triangle\alpha)}{m_{i,a_1+\triangle\alpha}} \\ &= \frac{1}{m_{i,\alpha}} \sum_{j\in R_{i,\alpha}} f_j + \mu_{i,\alpha}\frac{2\alpha}{m_{i,\alpha}}.\end{aligned}$$

On the other hand, we have by (A.16) that

$$u_i(\alpha) = \frac{1}{m_{i,\alpha}} \sum_{j\in R_{i,\alpha}} f_j + \mu_{i,\alpha}\frac{2\alpha}{m_{i,\alpha}}\,.$$

Thus, $u_i(\alpha) = \tilde{u}_i(\triangle\alpha)$.

Now one can continue the same way by considering $[a_2, a_3)$ and so on. Since there is only a finite number N of pixels and some of these pixels merge at the points a_j the process stops after a finite number of n steps with output $u(a_n)$ which by (A.16) reads as

$$u_i(a_n) = \frac{1}{N}\sum_{k=0}^{N-1} f_k$$

for all $i = 0, \ldots, N-1$. This completes the proof. □

B Notation

I	Input gray value image
$\mathbf{I} = (I_1, ..., I_M)$	Input multi-channel image
R, G, B	RGB color channels of input image $\mathbf{I} = (R, G, B)$
x, y	Spatial coordinates in an image or image sequence
z	Temporal coordinate in an image sequence
$\mathbf{x}$	Coordinate vector, i.e., either $\mathbf{x} = (x, y)$ in images or $\mathbf{x} = (x, y, z)$ in image sequences
t	Evolution time in evolution equations
Ω	Rectangular image domain, $\Omega \subset \mathbb{R}^2$ for images, $\Omega \subset \mathbb{R}^3$ for image sequences
∂_a	Abbreviation for $\frac{\partial}{\partial a}$
a_b	Abbreviation for $\frac{\partial a}{\partial b}$, if not explicitly defined otherwise
a_{bc}	Abbreviation for $\frac{\partial a}{\partial b \partial c}$
∇a	Gradient of a, i.e., $(\partial_x, \partial_y)^\top a$ in the spatial case and $(\partial_x, \partial_y, \partial_z)^\top a$ in the spatio-temporal case
Δa	Laplacian of a, i.e., $\partial_x^2 a + \partial_y^2 a$ in the spatial case and $\partial_x^2 a + \partial_y^2 a + \partial_z^2 a$ in the spatio-temporal case
$\mathrm{div}\,(\mathbf{a})$	$\partial_x a_1 + \partial_y a_2$ in the spatial case $\partial_x a_1 + \partial_y a_2 + \partial_z a_3$ in the spatio-temporal case
$\lvert\mathbf{a}\rvert$	Magnitude of a vector $\mathbf{a}$
M	Number of color or feature channels
N	Number of regions
u	Evolving data due to diffusion or energy minimization, in optic flow estimation u has a special meaning (see next item)
$\mathbf{w} = (u, v, 1)$	Optic flow vector
$\mathbf{u} = (u_1, ..., u_M)$	Evolving vector-valued data due to diffusion or energy minimization
K_ρ	Gaussian kernel with standard deviation ρ

$a * b$	Convolution of a with b
g	Diffusivity
λ	In the scope of diffusion: contrast parameter, in the scope of image segmentation with shape knowledge: weighting parameter for the shape constraint
ϵ	Some small positive constant
p	Parameter for edge preservation/enhancement, $p \in \mathbb{R}, p \geq 0$
h	Spatial grid size in numerical schemes
h_a	Spatial grid size in a-direction
τ	Time step size in numerical schemes
$a_{i,j}^k$	Discretized variable a at pixel (i, j) after iteration k
$\mathcal{N}(i)$	Set of pixels in the neighborhood of pixel i, 4 neighbors in 2-D, 6 neighbors in 3-D
$\|\mathcal{N}(i)\|$	Number of pixels in the neighborhood of pixel i
J_0	Unsmoothed structure tensor
J_ρ	Linear structure tensor, i.e. J_0 smoothed with a Gaussian kernel with standard deviation ρ
J_t	Nonlinear structure tensor obtained by smoothing the second moment matrix with nonlinear diffusion with diffusion time t
$\Psi(s)$	Penalizer function in variational methods, e.g., the quadratic error norm $\Psi(s) = s^2$
$\Psi'(s^2)$	Derivative of Ψ with respect to s^2
α	Weighting parameter for the smoothness constraint
γ	Weighting parameter for the gradient constancy constraint in optic flow computation
ω	Relaxation parameter in successive over-relaxation method for solving linear systems
$\Re(a)$	Real part of a complex value a
$\Im(a)$	Imaginary part of a complex value a
m	Local scale
η	Scaling factor between successive resolution levels, $\eta \in (0, 1)$
Ω_i	One of N regions, $\Omega_i \subset \Omega$
Γ_i	Boundary of region Ω_i
$C(s, t)$	Curve with arc-length parametrization at evolution time t
Φ	Level set function

$H(s)$ Regularized Heaviside function $H(s) = \mathrm{erf}(s)$, i.e., integral of the Gaussian function $K_1 = \frac{1}{\sqrt{2\pi}} e^{-\frac{s^2}{2}}$

$H'(s)$ Derivative of the Heaviside function with respect to s, i.e. $H'(s) = K_1$

$\mathbf{1}_{\{a=b\}}$ Indicator function defined as $\mathbf{1}_{\{a=b\}} := \begin{cases} 1 & \text{if } a = b \\ 0 & \text{else} \end{cases}$

$p_i(s)$ Probability density of a value s in region Ω_i

μ Mean value

σ Standard deviation

ν Weighting parameter for the length constraint in image segmentation

ϱ Weighting parameter for penalizing the number of regions in multi-region segmentation

P Projection matrix that projects 3-D points onto the image plane of the camera

$\mathbf{X}$ Homogeneously scaled 3-D point $(x_1, x_2, x_3, 1)^\top$

$\theta\xi$ 3-D pose parameters describing a screw motion, i.e., a rigid motion consisting of a translation and a rotation in 3-D space

C Contributions and Publications

Contributions

This work contains several novel contributions on PDE based image analysis, including the fields of *nonlinear diffusion, texture, optic flow,* and *image segmentation.* In addition, application of image processing techniques to hyperbolic conservation laws have been investigated. In detail, the following novelties have been presented:

- The special image denoising methods of total variation flow and total variation regularization have been analyzed theoretically. As a result, analytical solutions for both processes can be found in the discrete one-dimensional case. Further on, it can be shown that both methods are equivalent in this case. These results have been published in [BWSW03] and [SWB$^+$04].

- A class of diffusivity functions which leads to unbounded diffusivities and includes the diffusivity for total variation flow has been investigated. It has been applied to smoothing tasks in the scope of optic flow estimation and feature smoothing in image segmentation. The application to feature smoothing has been published in [BRDW03a] and [BRDW03b].

- Numerics for diffusion with unbounded diffusivities has been analyzed. The suggested discretization scheme for nonlinear diffusion leads to more accurate approximations of the continuous diffusion equation, in particular in case of TV flow. Furthermore, a new iteration scheme based on the mass exchange between two pixels has been investigated. It has been published in [SWB$^+$04]. A comparison of several numerical schemes for the solution of diffusion equations has revealed their pros and cons.

- The so-called *structure tensor*, a tool for orientation estimation, has been improved by replacing Gaussian smoothing by total variation flow. This *nonlinear structure tensor* has been published in [BWBM04]. Its application to texture segmentation and a review on adaptive structure tensors as well as their applications in general have been published in [RBD03] and [BvdBL$^+$05], respectively.

- Considering numerical schemes for approximating hyperbolic conservation laws in fluid dynamics, PDE based image processing techniques can be useful to deal with the occurring oscillation or blurring artifacts. According to this principle, TV flow has been employed in order to remove oscillations created by the Lax-Wendroff scheme. A stabilized inverse diffusion process, on the other hand, helps to improve the shock resolution of monotone schemes. In particular, the monotone Upwind scheme has been extended by a nonlinear inverse diffusion process. The resulting scheme yields favorable results. It has been published in [BBSW05].

- A local, region based scale measure has been suggested on the basis of TV flow. It is applied in order to estimate the scale of texture elements for texture discrimination. This scale measure has been published in [BW04b].

- Combination of the structure tensor and the scale measure leads to a texture model which resembles the model of a Gabor filter bank. However, while a Gabor filter bank yields a high-dimensional redundant feature space, the dimensionality of the proposed features is significantly lower. The new feature space has been applied in the scope of texture segmentation.

- In the scope of variational optic flow estimation, a method using a gradient constancy assumption has been introduced, which yields robustness against illumination changes between the corresponding images. This method has been published in [BBPW04].

- While the linearization of the gray value constancy assumption is well-established in optic flow literature, this linearization becomes in the presence of larger displacements a quite bad approximation of the actually nonlinear equation. Therefore, it has been suggested to use the original, non-linearized equation for optic flow computation. A minimization scheme based on nested fixed point iterations has been introduced to tackle the nonlinearities. It has turned out that this minimization scheme is equivalent to an earlier proposed multi-resolution method based on so-called image warping. Thus the minimization scheme provides a theoretical justification for this so far purely algorithmically motivated method. This has been published in [BBPW04].

- A level set based segmentation framework that integrates texture and motion cues derived in Chapter 4 has been introduced. Different statistical models for describing the interior of regions have been suggested including a nonparametric model and a model for local statistics. In order to deal with local optima, which often appear in connection with image segmentation, a coarse-to-fine minimization strategy has been proposed. Large parts of this work has been published in [RBD03, BRDW03a, BRDW03b].

- The segmentation technique has originally been restricted to split an image into solely two regions: one object and the background. This restriction emanates from the representation of regions in the level set framework. Since this is an unacceptable constraint for images containing multiple objects, the level set framework has been extended to deal with arbitrary many regions by means of coupled evolution equations. Moreover, the optimum number of regions is determined by a hierarchical splitting approach based on the conventional two-region segmentation scheme. This technique for multiple region segmentation has been published in [BW04a].

- As long as image segmentation techniques only make use of the data provided by the image, the quality of the results depends enormously on whether the objects in the image provide a cue that enables to distinguish them unambiguously from the rest of the image. In order to diminish this dependency, shape knowledge about the object has been integrated into segmentation. While this fundamental concept is not new, the novelty consists of the integration of *three-dimensional* shape knowledge which is projected onto the image plane. For this, its pose in three-dimensional space has to be estimated. Joint segmentation and pose estimation accomplishes both tasks at once and thereby improves the robustness of segmentation in scenes with ambiguous data. This has been joint work with Bodo Rosenhahn.

As indicated in the particular items, most novelties have been published previously on conferences or in scientific journals. Also note that many of these works have been in cooperation with other researchers. In such cases these researchers are either mentioned explicitly or they have been authors or co-authors of the respective articles.

Publications

1. T. Brox, D. Farin, and P. H. N. de With. Multi-stage region merging for image segmentation. In *Proc. 22nd Symposium on Information and Communication Theory in the Benelux*, pp. 189–196, Enschede, The Netherlands, May 2001.

2. T. Brox. Smoothing of matrix-valued data. Master's thesis, Department of Mathematics and Computer Science, University of Mannheim, Germany, May 2002.

3. T. Brox and J. Weickert. Nonlinear matrix diffusion for optic flow estimation. In L. Van Gool, editor, *Pattern Recognition*, volume 2449 of *Lecture Notes in Computer Science*, pages 446–453. Springer, Berlin, September 2002.

4. J. Weickert and T. Brox. Diffusion and regularization of vector- and matrix-valued images. In M. Z. Nashed and O. Scherzer, editors, *Inverse Problems, Image Analysis, and Medical Imaging*, volume 313 of *Contemporary Mathematics*, pages 251-268. AMS, Providence, 2002.

5. M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Proc. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 699-704, Madison, WI, June 2003. IEEE Computer Society Press.

6. T. Brox, M. Welk, G. Steidl, and J. Weickert. Equivalence results for TV diffusion and TV regularisation. In L. D. Griffin and M. Lillholm, editors, *Scale-Space Methods in Computer Vision*, volume 2695 of *Lecture Notes in Computer Science*, pages 86-100, Berlin, June 2003. Springer.

7. T. Brox, M. Rousson, R. Deriche, and J. Weickert. Unsupervised segmentation incorporating colour, texture, and motion. In N. Petkov and M. A. Westenberg, editors, *Computer Analysis of Images and Patterns*, volume 2756 of *Lecture Notes in Computer Science*, pages 353-360. Springer, Berlin, August 2003.

8. J. Weickert, A. Bruhn, N. Papenberg, and T. Brox. Variational optic flow computation: from continuous models to algorithms. In L. Alvarez, editor, *Proc. IWCVIA '03: International Workshop on Computer Vision and Image Analysis*, pages 1–6, University of Las Palmas de Gran Canaria, Spain, February 2004.

9. G. Steidl, J. Weickert, T. Brox, P. Mrázek, and M. Welk. On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and SIDEs. *SIAM Journal on Numerical Analysis*, 42(2):686-713, May 2004.

10. T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *Computer Vision - Proc. 8th European Conference on Computer Vision*, volume 3024 of *Lecture Notes in Computer Science*, pages 25-36. Springer, Berlin, May 2004. **Received *The Longuet-Higgins Best Paper Award.***

11. T. Brox and J. Weickert. A TV flow based local scale measure for texture discrimination. In T. Pajdla and J. Matas, editors, *Computer Vision - Proc. 8th European Conference on Computer Vision*, volume 3022 of *Lecture Notes in Computer Science*, pages 578–590. Springer, Berlin, May 2004.

12. T. Brox and J. Weickert. Level set based segmentation of multiple objects. In C. E. Rasmussen, H. H. Bülthoff, M. A. Giese, and B. Schölkopf, editors, *Pattern Recognition*, volume 3175 of *Lecture Notes in Computer Science*, pages 415-423. Springer, August 2004.

13. M. Breuß, T. Brox, T. Sonar, and J. Weickert. Stabilised nonlinear inverse diffusion for approximating hyperbolic PDEs. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*, pages 536-547. Springer, Berlin, April 2005.

14. M. Welk, D. Theis, T. Brox, and J. Weickert. PDE-based deconvolution with forward-backward diffusivities and diffusion tensors. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*, pages 585-597. Springer, Berlin, April 2005.

15. J. Weickert, C. Feddern, M. Welk, B. Burgeth, and T. Brox. PDEs for tensor image processing. In J. Weickert and H. Hagen, editors, *Visualization and Processing of Tensor Fields.* Springer, Berlin, 2005. To appear.

16. T. Brox, R. van den Boomgaard, F. B. Lauze, J. van de Weijer, J. Weickert, P. Mrázek, and P. Kornprobst. Adaptive structure tensors and their applications. In J. Weickert and H. Hagen, editors, *Visualization and Processing of Tensor Fields.* Springer, Berlin, 2005. To appear.

17. N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. International Journal of Computer Vision, 2005. To appear.

Bibliography

[AB94] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6), June 1994.

[ABCM01] F. Andreu, C. Ballester, V. Caselles, and J. M. Mazón. Minimizing total variation flow. *Differential and Integral Equations*, 14(3):321–360, March 2001.

[ACDM02] F. Andreu, V. Caselles, J. I. Diaz, and J. M. Mazón. Qualitative properties of the total variation flow. *Journal of Functional Analysis*, 188(2):516–547, February 2002.

[ADK99] G. Aubert, R. Deriche, and P. Kornprobst. Computing optical flow via variational techniques. *SIAM Journal on Applied Mathematics*, 60(1):156–182, 1999.

[ADPS02] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sánchez. Symmetrical dense optical flow estimation with occlusions detection. In *Proc. 7th European Conference on Computer Vision*, volume 2350 of *Lecture Notes in Computer Science*, pages 721–736. Springer, Berlin, 2002.

[ADSW02] L. Alvarez, R. Deriche, J. Sánchez, and J. Weickert. Dense disparity map estimation respecting image derivatives: a PDE and scale-space based approach. *Journal of Visual Communication and Image Representation*, 13(1/2):3–21, 2002.

[AELS99] L. Alvarez, J. Esclarín, M. Lefébure, and J. Sánchez. A PDE model for computing the optical flow. In *Proc. XVI Congreso de Ecuaciones Diferenciales y Aplicaciones*, pages 1349–1356, Las Palmas de Gran Canaria, Spain, September 1999.

[AGLM93] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel. Axioms and fundamental equations in image processing. *Archive for Rational Mechanics and Analysis*, 123:199–257, 1993.

[Ana89] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[AS95] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118(2):269–277, 1995.

[AT92] L. Ambrosio and V. Tortorelli. Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence. *Bolletino della Unione Matematica Italiana*, 7:105–123, 1992.

[AV94] R. Acar and C. R. Vogel. Analysis of bounded variation penalty methods for ill–posed problems. *Inverse Problems*, 10:1217–1229, 1994.

[AWS00] L. Alvarez, J. Weickert, and J. Sánchez. Reliable estimation of dense optical flow fields with large displacements. *International Journal of Computer Vision*, 39(1):41–56, August 2000.

[BA91] M. J. Black and P. Anandan. Robust dynamic motion estimation over time. In *Proc. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 292–302, Maui, HI, June 1991. IEEE Computer Society Press.

[BA96] M. J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, January 1996.

[BB73] J. P. Boris and D. L. Book. Flux corrected transport I: Shasta, a fluid transport algorithm that works. *Journal of Computational Physics*, 11(1):38–69, 1973.

[BB76] J. P. Boris and D. L. Book. Flux corrected transport III: Minimal-error FCT algorithms. *Journal of Computational Physics*, 20:397–431, 1976.

[BBH75] D. L. Book, J. P. Boris, and K. Hain. Flux corrected transport II: Generalizations of the method. *Journal of Computational Physics*, 18:248–283, 1975.

[BBPW04] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *Computer Vision - Proc. 8th European Conference on Computer Vision*, volume 3024 of *Lecture Notes in Computer Science*, pages 25–36. Springer, Berlin, May 2004.

[BBSW05] M. Breuß, T. Brox, T. Sonar, and J. Weickert. Stabilised nonlinear inverse diffusion for approximating hyperbolic PDEs. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale Space and PDE Methods in Computer Vision*, number 3459 in Lecture Notes in Computer Science, pages 536–547. Springer, Berlin, April 2005.

[BCG90] A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, January 1990.

[BCN02] G. Bellettini, V. Caselles, and M. Novaga. The total variation flow in R^N. *Journal of Differential Equations*, 184(2):475–525, 2002.

[BF93] P. Bouthemy and E. François. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *International Journal of Computer Vision*, 10(2):157–182, April 1993.

[BFB94] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, February 1994.

[BGK$^+$89] J. Beveridge, J. Griffith, R. Kohler, A. Hanson, and E. Riseman. Segmenting images using localized histograms and region merging. *International Journal of Computer Vision*, 2(3):311–352, 1989.

[BGS02] A. Bürgel, T. Grahs, and T. Sonar. From continuous recovery to discrete filtering in numerical approximations of conservation laws. *Applied Numerical Mathematics*, 42:47–60, 2002.

[BGW91] J. Bigün, G. H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, August 1991.

[BHS98] A. Bab-Hadiashar and D. Suter. Robust optic flow computation. *International Journal of Computer Vision*, 29(1):59–77, August 1998.

[Bih95] D. Le Bihan. *Magnetic Resonance Imaging of Diffusion and Perfusion: Applications to Functional Imaging.* Lippincott-Raven Press, New York, 1995.

[BJ96] M. J. Black and A. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):972–986, October 1996.

[BJF$^+$04] A. Bruhn, T. Jakob, M. Fischer, T. Kohlberger, J. Weickert, U. Brüning, and C. Schnörr. High performance cluster computing with 3-D nonlinear diffusion filters. *Real-Time Imaging*, 10(1):41–51, 2004.

[BL79] S. Beucher and C. Lantuéjoul. Use of watersheds in contour detection. In *Proc. International Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation*, pages 2.1–2.12, Rennes, France, September 1979. IRISA Report No. 131.

[BM98] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 8–15, Santa Barbara, California, 1998.

[BMB94] P. J. Basser, J. Mattiello, and D. Le Bihan. MR diffusion tensor spectroscopy and imaging. *Biophysical Journal*, 66:259–267, 1994.

[BRDW03a] T. Brox, M. Rousson, R. Deriche, and J. Weickert. Unsupervised segmentation incorporating colour, texture, and motion. Technical Report 4760, INRIA Sophia-Antipolis, France, 2003.

[BRDW03b] T. Brox, M. Rousson, R. Deriche, and J. Weickert. Unsupervised segmentation incorporating colour, texture, and motion. In N. Petkov and M. A. Westenberg, editors, *Computer Analysis of Images and Patterns*, volume 2756 of *Lecture Notes in Computer Science*, pages 353–360. Springer, Berlin, August 2003.

[Bro66] P. Brodatz. *Textures: a Photographic Album for Artists and Designers.* Dover, New York, 1966.

[Bro02] T. Brox. Smoothing of matrix-valued data. Master's thesis, Department of Mathematics and Computer Science, University of Mannheim, Germany, May 2002.

[Bru05] A. Bruhn. *Variational Optic Flow Computation: Accurate Modelling and Efficient Numerics.* PhD thesis, Faculty of Mathematics and Computer Science, Saarland University, Germany, 2005. To appear.

[BSMH98] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3):421–432, March 1998.

[Bür05] A. Bürgel. *Nichtlineare, diskrete Filteralgorithmen zur numerischen Lösung hyperbolischer Erhaltungsgleichungen.* PhD thesis, Department of Mathmematics and Computer Science, Technical University of Braunschweig, Germany, 2005.

[BvdBL^{+}05] T. Brox, R. van den Boomgaard, F. B. Lauze, J. van de Weijer, J. Weickert, P. Mrázek, and P. Kornprobst. Adaptive structure tensors and their applications. In J. Weickert and H. Hagen, editors, *Visualization and Processing of Tensor Fields.* Springer, Berlin, 2005. To appear.

[BW02] T. Brox and J. Weickert. Nonlinear matrix diffusion for optic flow estimation. In L. Van Gool, editor, *Pattern Recognition*, volume 2449 of *Lecture Notes in Computer Science*, pages 446–453. Springer, Berlin, September 2002.

[BW04a] T. Brox and J. Weickert. Level set based segmentation of multiple objects. In C.E. Rasmussen, H.H. Bülthoff, M.A. Giese, and B. Schölkopf, editors, *Pattern Recognition*, volume 3175 of *Lecture Notes in Computer Science*, pages 415–423. Springer, August 2004.

[BW04b] T. Brox and J. Weickert. A TV flow based local scale measure for texture discrimination. In T. Pajdla and J. Matas, editors, *Computer Vision - Proc. 8th European Conference on Computer Vision*, volume 3022 of *Lecture Notes in Computer Science*, pages 578–590. Springer, Berlin, May 2004.

[BW05] A. Bruhn and J. Weickert. Confidence measures for variational optic flow methods. In R. Klette, R. Kozera, L. Noakes, and J. Weickert, editors, *Geometric Properties from Incomplete Data.* Kluwer, 2005. To appear.

[BWBM04] T. Brox, J. Weickert, B. Burgeth, and P. Mrázek. Nonlinear structure tensors. Technical Report 113, Dept. of Mathematics, Saarland University, Saarbrücken, Germany, October 2004.

[BWF^{+}03] A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnörr. Real-time optic flow computation with variational methods. In N. Petkov and M. A. Westenberg, editors, *Computer Analysis of Images and Patterns*, volume 2756 of *Lecture Notes in Computer Science*, pages 222–229. Springer, Berlin, 2003.

[BWF^{+}05] A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnörr. Variational optic flow computation in real-time. *IEEE Transactions on Image Processing*, May 2005. To appear.

[BWKS05] A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr. Discontinuity-preserving computation of variational optic flow in real-time. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*, pages 279–290. Springer, Berlin, April 2005.

[BWS05] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[BWSW03] T. Brox, M. Welk, G. Steidl, and J. Weickert. Equivalence results for TV diffusion and TV regularisation. In L. D. Griffin and M. Lillholm, editors, *Scale-Space Methods in Computer Vision*, volume 2695 of *Lecture Notes in Computer Science*, pages 86–100. Springer, Berlin, June 2003.

[BZ87] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, Cambridge, MA, 1987.

[Can86] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[CBA93] L. D. Cohen, E. Bardinet, and N. Ayache. Surface reconstruction using active contour models. In *SPIE Conference on Geometric Methods in Computer Vision*, San Diego, CA, July 1993.

[CBFAB97] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, 1997.

[CC96] V. Caselles and T. Coll. Snakes in movement. *SIAM Journal on Numerical Analysis*, 33:2445–2456, 1996.

[CCCD93] V. Caselles, F. Catté, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66:1–31, 1993.

[CDLL98] A. Chambolle, R. A. DeVore, N. Lee, and B. L. Lucier. Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transactions on Image Processing*, 7(3):319–335, March 1998.

[CFL28] R. Courant, K. O. Friedrichs, and H. Lewy. Über die partiellen Differentialgleichungen der mathematischen Physik. *Mathematische Annalen*, 100:32–74, 1928.

[CGGM98] C. K. Chu, I. Glad, F. Godtliebsen, and J. S. Marron. Edge-preserving smoothers for image processing. *Journal of the American Statistical Association*, 93(442):526–556, 1998.

[CH53] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 1. Interscience, New York, 1953.

[Cha04] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20:89–97, 2004.

[Che95] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[Cho93] D. Chopp. Computing minimal surfaces via level set curvature flow. *Journal of Computational Physics*, 106(1):77–91, May 1993.

[Cia78] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North-Holland, Amsterdam, 1978.

[CKS95] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proc. Fifth International Conference on Computer Vision*, pages 694–699, Cambridge, MA, June 1995. IEEE Computer Society Press.

[CKS97] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22:61–79, 1997.

[CKS03] D. Cremers, T. Kohlberger, and C. Schnörr. Shape statistics in kernel space for variational image segmentation. *Pattern Recognition*, 36(9):1929–1943, September 2003.

[CLMC92] F. Catté, P.-L. Lions, J.-M. Morel, and T. Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical Analysis*, 32:1895–1909, 1992.

[CM02] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[Coh91] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211–218, March 1991.

[Coh93] I. Cohen. Nonlinear variational method for optical flow computation. In *Proc. Eighth Scandinavian Conference on Image Analysis*, volume 1, pages 523–530, Tromsø, Norway, May 1993.

[COS01] T. F. Chan, S. Osher, and J. Shen. The digital TV filter and nonlinear denoising. *IEEE Transactions on Image Processing*, 10(2):231–241, February 2001.

[COS04] D. Cremers, S. Osher, and S. Soatto. A multi-modal translation-invariant shape prior for level set segmentation. In C.-E. Rasmussen, H. Bülthoff, M. Giese, and B. Schölkopf, editors, *Pattern Recognition*, volume 3175 of *Lecture Notes in Computer Science*, pages 36–44. Springer, Berlin, August 2004.

[Cre02] D. Cremers. *Statistical Shape Knowledge in Variational Image Segmentation.* PhD thesis, Department of Mathematics and Computer Science, University of Mannheim, Germany, July 2002.

[CS05] D. Cremers and S. Soatto. Motion competition: A variational framework for piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265, May 2005. To appear.

[CSS04] D. Cremers, N. Sochen, and C. Schnörr. Multiphase dynamic labeling for variational recognition-driven image segmentation. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conference on Computer Vision*, volume 3024 of *Lecture Notes in Computer Science*, pages 74–86. Springer, Berlin, May 2004.

[CSV00] T. Chan, B.Y. Sandberg, and L. Vese. Active contours without edges for vector-valued images. *Journal of Visual Communication and Image Representation*, 11:130–141, 2000.

[CSW01] D. Cremers, C. Schnörr, and J. Weickert. Diffusion-snakes: combining statistical shape knowledge and image information in a variational framework. In *Proc. First IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 137–144, Vancouver, Canada, July 2001. IEEE Computer Society Press.

[CTDF04] C. Chefd'Hotel, D. Tschumperlé, R. Deriche, and O. Faugeras. Regularizing flows for constrained matrix-valued images. *Journal of Mathematical Imaging and Vision*, 20(1/2):147–162, 2004.

[CTT$^+$01] Y. Chen, S. Thiruvenkadam, H. D. Tagare, F. Huang, D. Wilson, and E. A. Geisler. On the incorporation of shape priors into geometric active contours. In *Proc. First IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 145–152, Vancouver, Canada, July 2001. IEEE Computer Society Press.

[CTT$^+$02] Y. Chen, H. D. Tagare, S. Thiruvenkadam, F. Huang, K. S. Gopinath, R. W. Briggs, and E. A. Geisler. Using prior shapes in geometric active contours in a variational framework. *International Journal of Computer Vision*, 50(3):315–328, December 2002.

[CTWS02] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr. Diffusion snakes: introducing statistical shape knowledge into the mumford-shah functional. *International Journal of Computer Vision*, 50(3):295–313, December 2002.

[CV99] T. Chan and L. Vese. An active contour model without edges. In M. Nielsen, P. Johansen, O. F. Olsen, and J. Weickert, editors, *Scale-Space Theories in Computer Vision*, volume 1682 of *Lecture Notes in Computer Science*, pages 141–151. Springer, 1999.

[CV01] T. F. Chan and L. A. Vese. A level set algorithm for minimizing the Mumford–Shah functional in image processing. In *Proc. First IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 161–168, Vancouver, Canada, July 2001. IEEE Computer Society Press.

[Dau85] J. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, 2(7):1160–1169, 1985.

[DFdGtH04] R. Duits, L. Florack, J. de Graaf, and B. ter Haar Romeny. On the axioms of scale space theory. *Journal of Mathematical Imaging and Vision*, 20(3):267–298, May 2004.

[DHW94] D. Dunn, W. E. Higgins, and J. Wakeley. Texture segmentation using 2-D Gabor elementary functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):130–149, February 1994.

[Di 86] S. Di Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics and Image Processing*, 33:116–125, 1986.

[DJ94] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrica*, 81(3):425–455, 1994.

[DK00] F. Dibos and G. Koepfler. Global total variation minimization. *SIAM Journal on Numerical Analysis*, 37(2):646–664, 2000.

[Don95] D. L. Donoho. De-noising by soft thresholding. *IEEE Transactions on Information Theory*, 41:613–627, 1995.

[DT79] A. Dervieux and F. Thomasset. A finite element method for the simulation of Rayleigh–Taylor instability. In R. Rautman, editor, *Approximation Methods for Navier–Stokes Problems*, volume 771 of *Lecture Notes in Mathematics*, pages 145–158. Springer, Berlin, 1979.

[EF98] M. Elad and A. Feuer. Recursive optical flow estimation – adaptive filtering approach. *Journal of Visual Communication and Image Representation*, 9(2):119–138, June 1998.

[Els61] L. E. Elsgolc. *Calculus of Variations.* Pergamon, Oxford, 1961.

[ELS89] B. Engquist, P. Lötstedt, and B. Sjögreen. Nonlinear filters for efficient shock computation. *Mathematics of Computation*, 52(186):509–537, April 1989.

[ES91] L. C. Evans and J. Spruck. Motion of level sets by mean curvature I. *Journal of Differential Geometry*, 33:635–681, 1991.

[EZ98] J. H. Elder and S. W. Zucker. Local scale control for edge detection and blur estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):699–716, July 1998.

[Far00] G. Farnebäck. Fast and accurate motion estimation using orientation tensors and parametric motion models. In *Proc. 15th International Conference on Pattern Recognition*, volume 1, pages 135–139, Barcelona, Spain, September 2000.

[Far01] G. Farnebäck. Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field. In *Proc. Eighth International Conference on Computer Vision*, volume 1, pages 171–177, Vancouver, Canada, July 2001. IEEE Computer Society Press.

[Fau93] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint.* MIT Press, Cambridge, MA, 1993.

[FG87] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, Interlaken, Switzerland, June 1987.

[Fie87] D. J. Field. Relation between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4:2379–2394, 1987.

[Fil88] A. F. Filippov. *Differential Equations with Discontinuous Righthand Sides.* Kluwer, Dordrecht, 1988.

[FLP01] O. Faugeras, Q.-T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images.* MIT Press, Cambridge, MA, 2001.

[FP02] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach.* Prentice Hall, Upper Saddle River, 2002.

[FPT01] P. Fua, R. Plänkers, and D. Thalmann. Tracking and modeling people in video sequences. *Computer Vision and Image Understanding*, 81(3):285–302, March 2001.

[FS01] M. Felsberg and G. Sommer. Scale-adaptive filtering derived from the Laplace equation. In B. Radig and S. Florczyk, editors, *Pattern Recognition*, volume 2032 of *Lecture Notes in Computer Science*, pages 95–106. Springer, Berlin, 2001.

[Gab46] D. Gabor. Theory of communication. *The Journal of the Institution of Electrical Engineers*, 93:429–457, 1946.

[Gag86] M. Gage. On area-preserving evolution equation for plane curves. *Contemporary Mathematics*, 23:51–62, 1986.

[GD96] D. Gavrila and L. Davis. 3D model based tracking of humans in action: a multiview approach. In *ARPA Image Understanding Workshop*, pages 73–80, Palm Springs, 1996.

[GF00] J. Gomes and O. Faugeras. Reconciling distance functions and level sets. *Journal of Visual Communication and Image Representation*, 11(2):209–223, 2000.

[GG84] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[GH86] M. Gage and R. S. Hamilton. The heat equation shrinking convex plane curves. *Journal of Differential Geometry*, 23:69–96, 1986.

[GKKJ92] G. Gerig, O. Kübler, R. Kikinis, and F. A. Jolesz. Nonlinear anisotropic filtering of MRI data. *IEEE Transactions on Medical Imaging*, 11:221–232, 1992.

[GKRR01] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Fast geodesic active contours. *IEEE Transactions on Image Processing*, 10(10):1467–1475, October 2001.

[GM04] B. Goldlücke and M. Magnor. Space-time isosurface evolution for temporally coherent 3D reconstruction. In *Proc. 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume I, pages 350–355, Washington D.C., July 2004.

[GMN$^+$98] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: an analysis of eight optical flow algorithms. In *Proc. 1998 British Machine Vision Conference*, Southampton, England, September 1998.

[GMS00] G. Gómez, J. L. Marroquín, and L. E. Sucar. Probabilistic estimation of local scale. In *Proc. International Conference on Pattern Recognition*, volume 3, pages 798–801, Barcelona, Spain, September 2000.

[GMS02] T. Grahs, A. Meister, and T. Sonar. Image processing for numerical approximations of conservation laws: nonlinear anisotropic artificial dissipation. *SIAM Journal on Scientific Computing*, 23(5):1439–1455, 2002.

[GR91] E. Godlewski and P.-A. Raviart. *Hyperbolic Systems of Conservation Laws.* Ellipses, Edition Marketing, 1991.

[Gra87] M. Grayson. The heat equation shrinks embedded plane curves to round points. *Journal of Differential Geometry*, 26:285–314, 1987.

[GS02] T. Grahs and T. Sonar. Entropy-controlled artificial anisotropic diffusion for the numerical solution of conservation laws based on algorithms from image processing. 13:176–194, 2002.

[GSBB03] M. Galun, E. Sharon, R. Basri, and A. Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. In *Proc. 9th IEEE International Conference on Computer Vision*, pages 716–723, Nice, France, October 2003.

[GSZ02] G. Gilboa, N. A. Sochen, and Y. Y. Zeevi. Forward-and-backward diffusion processes for adaptive image enhancement and denoising. *IEEE Transactions on Image Processing*, 11(7):689–703, 2002.

[Har87] A. Harten. ENO schemes with subcell resolution. *Journal of Computational Physics*, 83:148–184, 1987.

[HB97] T. Hofmann and J. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):1–14, January 1997.

[HOEC86] A. Harten, S. Osher, B. Engquist, and S. Chakravarthy. Some results on uniformly high-order accurate essentially nonoscillatory schemes. *Applied Numerical Mathematics*, 2:347–377, 1986.

[HPB98] T. Hofmann, J. Puzicha, and J. Buhmann. Unsupervised texture segmentation in a deterministic annealing framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):803–818, August 1998.

[HRRS86] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions.* MIT Press, Cambridge, MA, 1986.

[HS81] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[HS88] C. G. Harris and M. Stephens. A combined corner and edge detector. In *Proc. Fouth Alvey Vision Conference*, pages 147–152, Manchester, England, August 1988.

[HSSW02] W. Hinterberger, O. Scherzer, C. Schnörr, and J. Weickert. Analysis of optical flow models in the framework of calculus of variations. *Numerical Functional Analysis and Optimization*, 23(1/2):69–89, May 2002.

[Hub81] P. J. Huber. *Robust Statistics.* Wiley, New York, 1981.

[Iij62] T. Iijima. Basic theory on normalization of pattern (in case of typical one-dimensional pattern). *Bulletin of the Electrotechnical Laboratory*, 26:368–388, 1962. In Japanese.

[Iij63] T. Iijima. Theory of pattern recognition. *Electronics and Communications in Japan*, pages 123–134, November 1963. In English.

[Int71] International Commission on Illumination. Recommendations on uniform color spaces, color difference equations, psychometric color terms, 1971. Supplement No.2 to CIE publication No.15 (E.-1.3.1), TC-1.3.

[Jac96] P. T. Jackway. Gradient watersheds in morphological scale-space. *IEEE Transactions on Image Processing*, 5:913–921, 1996.

[JBJ96] S. Ju, M. Black, and A. Jepson. Skin and bones: multi-layer, locally affine, optical flow and regularization with transparency. In *Proc. 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 307–314, San Francisco, CA, June 1996. IEEE Computer Society Press.

[JK69] N. L. Johnson and S. Kotz. *Distributions in Statistics.* Houghton Mifflin Company, Boston, 1969.

[JK92] H. Jeong and I. Kim. Adaptive determination of filter scales for edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(5):579–585, May 1992.

[Jul81] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290:91–97, 1981.

[Jul86] B. Julesz. Texton gradients: the texton theory revisited. *Biological Cybernetics*, 54:245–251, 1986.

[KDA97] P. Kornprobst, R. Deriche, and G. Aubert. Nonlinear operators in image restoration. In *Proc. 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 325–330, San Juan, Puerto Rico, June 1997. IEEE Computer Society Press.

[Keu04] J. Keuchel. *Image Partitioning based on Semidefinite Programming.* PhD thesis, Department of Mathematics and Computer Science, University of Mannheim, Germany, September 2004.

[KFY$^+$02] J. Kim, J. Fisher, A. Yezzi, M. Cetin, and A. Willsky. Nonparametric methods for image segmentation using information theory and curve evolution. In *IEEE International Conference on Image Processing*, volume 3, pages 797–800, Rochester, NY, June 2002.

[KHS04] J. Keuchel, M. Heiler, and C. Schnörr. Hierarchical image segmentation based on semidefinite programming. In C.E. Rasmussen, H.H. Bülthoff, M.A. Giese, and B. Schölkopf, editors, *Pattern Recognition*, volume 3175 of *Lecture Notes in Computer Science*, pages 120–128. Springer, August 2004.

[Kic97] S. Kichenassamy. The Perona–Malik paradox. *SIAM Journal on Applied Mathematics*, 57:1343–1372, 1997.

[KKO$^+$95] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proc. Fifth International Conference on Computer Vision*, pages 810–815, Cambridge, MA, June 1995. IEEE Computer Society Press.

[KKO$^+$96] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Conformal curvature flows: from phase transitions to active vision. *Archive for Rational Mechanics and Analysis*, 134:275–301, 1996.

[KLM94] G. Koepfler, C. Lopez, and J. M. Morel. A multiscale algorithm for image segmentation by variational method. *SIAM Journal on Numerical Analysis*, 31(1):282–299, 1994.

[KS02] S. L. Keeling and R. Stollberger. Nonlinear anisotropic diffusion filters for wide range edge sharpening. *Inverse Problems*, 18:175–190, January 2002.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.

[LC01] M. Lefébure and L. D. Cohen. Image registration, optical flow and local rigidity. *Journal of Mathematical Imaging and Vision*, 14(2):131–147, March 2001.

[LCR03] H. Liu, R. Chellappa, and A. Rosenfeld. Accurate dense optical flow estimation using adaptive structure tensors and a parameteric model. *IEEE Transactions on Image Processing*, 12(10):1170–1180, October 2003.

[Lec89] Y. G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3:73–102, 1989.

[LeV92] R. J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhäuser, Basel, 1992.

[LGF00] M. E. Leventon, W. E. L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Proc. 2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 316–323, Hilton Head, SC, June 2000. IEEE Computer Society Press.

[Lin94] T. Lindeberg. *Scale-Space Theory in Computer Vision.* Kluwer, Boston, 1994.

[Lin99] T. Lindeberg. Principles for automatic scale selection. In B. Jähne, H. Haußecker, and P. Geißler, editors, *Handbook on Computer Vision and Applications*, volume 2, pages 239–274. Academic Press, Boston, USA, 1999.

[LK81] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Seventh International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, Canada, August 1981.

[LKL$^+$04] F. Lauze, P. Kornprobst, C. Lenglet, R. Deriche, and M. Nielsen. About some optical flow methods from structure tensors: review and contribution. In *RFIA 2004, Actes du* 14^e *Congrès Francophone AFRIF-AFIA*, volume 1, pages 283–292, Toulouse, January 2004. LAAS-CNRS. In French.

[LOC94] X.-D. Liu, S. Osher, and T. Chan. Weighted essentially non-osciallatory schemes. *Journal of Computational Physics*, 115:200–212, 1994.

[LV98] S.-H. Lai and B. C. Vemuri. Reliable and efficient computation of optical flow. *International Journal of Computer Vision*, 29(2):87–105, October 1998.

[LW60] P. D. Lax and B. Wendroff. Systems of conservation laws. *Communications on Pure and Applied Mathematics*, 13:217–237, 1960.

[Mac42] D. L. MacAdam. Visual sensitivities to color differences in daylight. *Journal of Optical Society of America*, 32:247–273, 1942.

[Mar80] S. Marcelja. Mathematical description of the response of simple cortical cells. *Journal of Optical Society of America*, 70:1297–1300, 1980.

[MBLS01] J. Malik, S. Belongie, T.K. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[MBO94] B. Merriman, J. Bence, and S. Osher. Motion of multiple junctions: A level set approach. *Journal of Computational Physics*, 112:334–363, 1994.

[MN01] M. Middendorf and H.-H. Nagel. Estimation and interpretation of discontinuities in optical flow fields. In *Proc. Eighth International Conference on Computer Vision*, volume 1, pages 178–183, Vancouver, Canada, July 2001. IEEE Computer Society Press.

[MN02] M. Middendorf and H.-H. Nagel. Empirically convergent adaptive estimation of grayvalue structure tensors. In L. van Gool, editor, *Proc. 24th DAGM Symposium*, volume 2449 of *Lecture Notes in Computer Science*, pages 66–74, Zürich, Switzerland, September 2002. Springer.

[MP96] E. Mémin and P. Pérez. Robust discontinuity-preserving model for estimating optical flow. In *Proc. 13th International Conference on Pattern Recognition*, pages 920–924, Vienna, Austria, August 1996.

[MP98a] E. Mémin and P. Pérez. Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE Transactions on Image Processing*, 7(5):703–719, May 1998.

[MP98b] E. Mémin and P. Pérez. A multigrid approach for hierarchical motion estimation. In *Proc. Sixth International Conference on Computer Vision*, pages 933–938, Bombay, India, January 1998. Narosa Publishing House.

[MP02] E. Mémin and P. Pérez. Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision*, 46(2):129–155, 2002.

[MS85] D. Mumford and J. Shah. Boundary detection by minimizing functionals, I. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 22–26, San Francisco, CA, June 1985. IEEE Computer Society Press.

[MS89] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.

[MS94] J.-M. Morel and S. Solimini. *Variational Methods in Image Segmentation.* Birkhäuser, Basel, 1994.

[MSV95] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, February 1995.

[MvdG97] E. Mammen and S. van de Geer. Locally adaptive regression splines. *Annals of Statistics*, 25(1):387–413, 1997.

[MWB04] P. Mrázek, J. Weickert, and A. Bruhn. On robust estimation and smoothing with spatial and tonal kernels. Technical Report 51, Series SPP-1114, Department of Mathematics, University of Bremen, Germany, June 2004.

[MWS03] P. Mrázek, J. Weickert, and G. Steidl. Correspondences between wavelet shrinkage and nonlinear diffusion. In L. D. Griffin and M. Lillholm, editors, *Scale-Space Methods in Computer Vision*, volume 2695 of *Lecture Notes in Computer Science*, pages 101–116. Springer, Berlin, 2003.

[Nag83] H.-H. Nagel. Constraints for the estimation of displacement vector fields from image sequences. In *Proc. Eighth International Joint Conference on Artificial Intelligence*, volume 2, pages 945–951, Karlsruhe, West Germany, August 1983.

[Nag87] H.-H. Nagel. On the estimation of optical flow: relations between different approaches and some new results. *Artificial Intelligence*, 33:299–324, 1987.

[Nag90] H.-H. Nagel. Extending the 'oriented smoothness constraint' into the temporal domain and the estimation of derivatives of optical flow. In O. Faugeras, editor, *Computer Vision – ECCV '90*, volume 427 of *Lecture Notes in Computer Science*, pages 139–148. Springer, Berlin, 1990.

[NE86] H.-H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:565–593, 1986.

[NFD97] M. Nielsen, L. Florack, and R. Deriche. Regularization, scale-space and edge detection filters. *Journal of Mathematical Imaging and Vision*, 7:291–307, 1997.

[NG98] H.-H. Nagel and A. Gehrke. Spatiotemporally adaptive estimation and segmentation of OF-fields. In H. Burkhardt and B. Neumann, editors, *Computer Vision – ECCV '98*, volume 1407 of *Lecture Notes in Computer Science*, pages 86–102. Springer, Berlin, 1998.

[OR90] S. Osher and L. I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis*, 27:919–940, 1990.

[OR91] S. Osher and L. I. Rudin. Shocks and other nonlinear filtering applied to image processing. In A. G. Tescher, editor, *Applications of Digital Image Processing XIV*, volume 1567 of *Proceedings of SPIE*, pages 414–431. SPIE Press, Bellingham, 1991.

[OS88] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[Par62] E. Parzen. On the estimation of a probability density function and the mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[PD99] N. Paragios and R. Deriche. Unifying boundary and region-based information for geodesic active tracking. In *Proc. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 300–305, Forth Collins, Colorado, June 1999.

[PD00a] N. Paragios and R. Deriche. Coupled geodesic active regions for image segmentation: A level set approach. In *Proc. 6th European Conference on Computer Vision*, number 2, pages 224–240, Dublin, Ireland, 2000.

[PD00b] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, March 2000.

[PD02a] N. Paragios and R. Deriche. Geodesic active regions: A new paradigm to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation*, 13(1/2):249–268, 2002.

[PD02b] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3):223–247, March 2002.

[PD05] N. Paragios and R. Deriche. Geodesic active regions and level set methods for motion estimation and tracking. *Computer Vision and Image Understanding*, 97(3):259–282, March 2005.

[PM90] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.

[PMO$^+$99] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local level set method. *Journal of Computational Physics*, 155(2):410–438, 1999.

[PNNT96] J. Portilla, R. Navarro, O. Nestares, and A. Tabernero. Texture synthesis-by-analysis based on a multiscale early-vision model. *Optical Engineering*, 35:2403–2417, 1996.

[Pot75] J. L. Potter. Velocity as a cue to segmentation. *IEEE Transactions on Systems, Man and Cybernetics*, 5:390–394, 1975.

[PRR03] N. Paragios, M. Rousson, and V. Ramesh. Distance tranforms for non-rigid registation. *Computer Vision and Image Understanding*, 23:142–165, 2003.

[PTVF92] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C.* Cambridge University Press, Cambridge, UK, second edition, 1992.

[PWH05] I. Pollak, A. S. Willsky, and Y. Huang. Nonlinear evolution equations as fast and exact solvers of estimation problems. *IEEE Transactions on Signal Processing*, 53(2):484–498, February 2005.

[PWK00] I. Pollak, A. S. Willsky, and H. Krim. Image segmentation and edge enhancement with stabilized inverse diffusion equations. *IEEE Transactions on Image Processing*, 9(2):256–266, February 2000.

[RBD03] M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Proc. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 699–704, Madison, WI, June 2003. IEEE Computer Society Press.

[RD96] L. Robert and R. Deriche. Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities. In B. Buxton and R. Cipolla, editors, *Computer Vision – ECCV '96*, volume 1064 of *Lecture Notes in Computer Science*, pages 439–451. Springer, Berlin, 1996.

[RD03] M. Rousson and R. Deriche. Dynamic segmentation of vector valued images. In S. Osher and N. Paragios, editors, *Geometric Level Set Methods in Imaging, Vision and Graphics.* Springer, August 2003.

[RdB93] T. R. Reed and J. M. H. du Buf. A review of recent texture segmentation and feature extraction techniques. *Computer Vision, Graphics and Image Processing*, 57(3):359–372, May 1993.

[RKS04] B. Rosenhahn, R. Klette, and G. Sommer. Silouette based human motion estimation. In C.E. Rasmussen, H.H. Bülthoff, M.A. Giese, and B. Schölkopf, editors, *Pattern Recognition*, volume 3175 of *Lecture Notes in Computer Science*, pages 294–301. Springer, August 2004.

[RM94] R. D. Richtmyer and K. W. Morton. *Difference Methods for Initial-Value Problems.* Krieger, Malabar, second edition, 1994.

[ROF92] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[Ron94] R. Ronfard. Region based strategies for active contour models. *International Journal of Computer Vision*, 13(2):229–251, October 1994.

[Ros56] F. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, 27:832–837, 1956.

[Ros03] B. Rosenhahn. *Pose Estimation Revisited.* PhD thesis, University of Kiel, Germany, September 2003.

[RP02] M. Rousson and N. Paragios. Shape priors for level set representations. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Computer Vision – ECCV 2002*, volume 2351 of *Lecture Notes in Computer Science*, pages 78–92. Springer, Berlin, 2002.

[RPD04] M. Rousson, N. Paragios, and R. Deriche. Implicit active shape models for 3D segmentation in MR imaging. In *7th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 3216 of *Lecture Notes in Computer Science*, pages 209–216. Springer, Berlin, September 2004.

[RPS05] B. Rosenhahn, C. Perwass, and G. Sommer. Pose estimation of free-form contours. *International Journal of Computer Vision*, 62(3):267–289, 2005. To appear.

[RRKS04] T. Riklin-Raviv, N. Kiryati, and N. Sochen. Unlevel-sets: geometry and prior-based segmentation. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conference on Computer Vision*, volume 3024 of *Lecture Notes in Computer Science*, pages 50–61. Springer, Berlin, May 2004.

[RS91] A. R. Rao and B. G. Schunck. Computing oriented texture fields. *CVGIP: Graphical Models and Image Processing*, 53:157–185, 1991.

[RS04] B. Rosenhahn and G. Sommer. Pose estimation of free-form objects. In T. Pajdla and J. Matas, editors, *Computer Vision - Proc. 8th European Conference on Computer Vision*, volume 3021 of *Lecture Notes in Computer Science*, pages 414–427. Springer, Berlin, May 2004.

[Rud87] L. I. Rudin. *Images, Numerical Analysis of Singularities and Shock Filters.* PhD thesis, California Institute of Technology, Pasadena, CA, 1987.

[SBB00] E. Sharon, A. Brandt, and R. Basri. Fast multiscale image segmentation. In *Proc. 2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 70–77, Hilton Head Island, South Carolina, June 2000.

[SBB01] E. Sharon, A. Brandt, and R. Basri. Segmentation and boundary detection using multiscale intensity measurements. In *Proc. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 469–476, Kauai, Hawaii, December 2001.

[SBFAZ00] C. Samson, L. Blanc-Féraud, G. Aubert, and J. Zerubia. A level set model for image classification. *International Journal of Computer Vision*, 40(3):187–197, 2000.

[Sch91] C. Schnörr. Determining optical flow for irregular domains by minimizing quadratic functionals of a certain class. *International Journal of Computer Vision*, 6(1):25–38, April 1991.

[Sch93] C. Schnörr. On functionals with greyvalue-controlled smoothness terms for determining optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1074–1079, 1993.

[Sch94a] C. Schnörr. Bewegungssegmentation von Bildfolgen durch die Minimierung konvexer nicht-quadratischer Funktionale. In W. Kropatsch and H. Bischof, editors, *Mustererkennung 1994*, pages 178–185. Springer, Berlin, 1994.

[Sch94b] C. Schnörr. Segmentation of visual motion by minimizing convex non-quadratic functionals. In *Proc. Twelfth International Conference on Pattern Recognition*, volume A, pages 661–663, Jerusalem, Israel, October 1994. IEEE Computer Society Press.

[Sch94c] C. Schnörr. Unique reconstruction of piecewise smooth images by minimizing strictly convex non-quadratic functionals. *Journal of Mathematical Imaging and Vision*, 4:189–198, 1994.

[SCT99] J. Sporring, C. I. Colios, and P. E. Trahanias. Generalized scale-selection. Technical Report 264, Foundation for Research and Technology - Hellas, Crete, Greece, December 1999.

[Set90] J. A. Sethian. Numerical algorithms for propagating interfaces: Hamilton-Jacobi equations and conservation laws. *Journal of Differential Geometry*, 31:131–161, 1990.

[Sil86] B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, New York, 1986.

[SKB01] N. Sochen, R. Kimmel, and F. Bruckstein. Diffusions and confusions in signal and image processing. *Journal of Mathematical Imaging and Vision*, 14(3):195–210, May 2001.

[SM97] J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proc. 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 731–737, San Juan, Puerto Rico, June 1997.

[SM00] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.

[Sny91] M. A. Snyder. On the mathematical foundations of smoothness constraints for the determination of optical flow and for surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:1105–1114, 1991.

[Son97] T. Sonar. *Mehrdimensionale ENO-Verfahren.* Advances in Numerical Mathematics. Teubner, Stuttgart, 1997.

[SSZ01] C. Sagiv, N. A. Sochen, and Y. Y. Zeevi. Geodesic active contours applied to texture feature space. In M. Kerckhove, editor, *Scale-Space and Morphology in Computer Vision*, volume 2106 of *Lecture Notes in Computer Science*, pages 344–352. Springer, Berlin, 2001.

[ST93] G. Sapiro and A. Tannenbaum. Affine invariant scale-space. *International Journal of Computer Vision*, 11:25–44, 1993.

[ST03] C. Sminchisescu and B. Triggs. Estimating articulated human motion with covariance scaled sampling. 22(6):371–391, 2003.

[Str97] D. M. Strong. *Adaptive Total Variation Minimizing Image Restoration.* PhD thesis, Department of Mathematics, University of California, Los Angeles, CA, 1997.

[SW99] J. Sporring and J. Weickert. Information measures in scale-spaces. *IEEE Transactions on Information Theory*, 45(3):1051–1058, April 1999.

[SW00] O. Scherzer and J. Weickert. Relations between regularization and diffusion filtering. *Journal of Mathematical Imaging and Vision*, 12(1):43–63, February 2000.

[SW02] G. Steidl and J. Weickert. Relations between soft wavelet shrinkage and total variation denoising. In L. Van Gool, editor, *Pattern Recognition*, volume 2449 of *Lecture Notes in Computer Science*, pages 198–205. Springer, Berlin, 2002.

[SWB$^+$04] G. Steidl, J. Weickert, T. Brox, P. Mrázek, and M. Welk. On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and SIDEs. *SIAM Journal on Numerical Analysis*, 42(2):686–713, May 2004.

[Tai92] X.-C. Tai. Global extrapolation with a parallel splitting method. *Numerical Algorithms*, 3(4):427–440, 1992.

[TD01] D. Tschumperlé and R. Deriche. Diffusion tensor regularization with contraints preservation. In *Proc. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 948–953, Kauai, HI, December 2001. IEEE Computer Society Press.

[TD03] D. Tschumperlé and R. Deriche. Variational frameworks for DT-MRI estimation, regularization and visualization. In *2nd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, pages 116–121, October 2003.

[TD05] D. Tschumperlé and R. Deriche. Vector-valued image regularization with PDE's: a common framework for different applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):506–517, April 2005.

[Tho49] L. H. Thomas. Elliptic problems in linear difference equations over a network. Technical report, Watson Scientific Computing Laboratory, Columbia University, New York, NJ, 1949.

[Tho80] W. B. Thompson. Combining motion and contrast for segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):543–549, 1980.

[Tho99] I. Thomas. Anisotropic adaptation and structure detection. Technical Report F11, Institute for Applied Mathematics, University of Hamburg, Germany, August 1999.

[Tik63] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Mathematics Doklady*, 4:1035–1038, 1963.

[Tis94] M. Tistarelli. Multiple constraints for optical flow. In J.-O. Eklundh, editor, *Computer Vision – ECCV '94*, volume 800 of *Lecture Notes in Computer Science*, pages 61–70. Springer, Berlin, 1994.

[TM98] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. Sixth International Conference on Computer Vision*, pages 839–846, Bombay, India, January 1998. Narosa Publishing House.

[Tsu00] V. I. Tsurkov. An analytical model of edge protection under noise suppression by anisotropic diffusion. *Journal of Computer and Systems Sciences International*, 39(3):437–440, 2000.

[TYW01] A. Tsai, A. Yezzi, and A. Willsky. Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Transactions on Image Processing*, 10(8):1169–1186, 2001.

[UGVT88] S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biological Cybernetics*, 60:79–87, 1988.

[VC02] L. Vese and T. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3):271–293, December 2002.

[vdBvdW02] R. van den Boomgaard and J. van de Weijer. Robust estimation of orientation for texture analysis. In *Proc. Texture 2002, 2nd International Workshop on Texture Analysis and Synthesis*, Copenhagen, June 2002.

[vHV91] S. van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis.* SIAM, Philadelphia, 1991.

[Vog02] C. R. Vogel. *Computational Methods for Inverse Problems.* SIAM, Philadelphia, 2002.

[WAHM99] G. Winkler, V. Aurich, K. Hahn, and A. Martin. Noise reduction in images: some recent edge-preserving methods. *Pattern Recognition and Image Analysis*, 9(4):749–766, 1999.

[WB97] J. Weickert and B. Benhamouda. A semidiscrete nonlinear scale-space theory and its relation to the Perona–Malik paradox. In F. Solina, W. G. Kropatsch, R. Klette, and R. Bajcsy, editors, *Advances in Computer Vision*, pages 1–10. Springer, Wien, 1997.

[WB02] J. Weickert and T. Brox. Diffusion and regularization of vector- and matrix-valued images. In M. Z. Nashed and O. Scherzer, editors, *Inverse Problems, Image Analysis, and Medical Imaging*, volume 313 of *Contemporary Mathematics*, pages 251–268. AMS, Providence, 2002.

[Wei98a] J. Weickert. *Anisotropic Diffusion in Image Processing.* Teubner, Stuttgart, 1998.

[Wei98b] J. Weickert. On discontinuity-preserving optic flow. In S. Orphanoudakis, P. Trahanias, J. Crowley, and N. Katevas, editors, *Proc. Computer Vision and Mobile Robotics Workshop*, pages 115–122, Santorini, Greece, September 1998.

[Wei99a] J. Weickert. Coherence-enhancing diffusion filtering. *International Journal of Computer Vision*, 31(2/3):111–127, April 1999.

[Wei99b] J. Weickert. Coherence-enhancing diffusion of colour images. *Image and Vision Computing*, 17(3/4):199–210, March 1999.

[Wei01] J. Weickert. Applications of nonlinear diffusion in image processing and computer vision. *Acta Mathematica Universitatis Comenianae*, 70(1):33–50, 2001.

[Wei02] G. W. Wei. Shock capturing by anisotropic diffusion oscillation reduction. *Computational Physics Communications*, 144:317–342, 2002.

[WH05] J. Weickert and H. Hagen. *Visualization and Processing of Tensor Fields.* Springer, Berlin, 2005. To appear.

[WHS$^+$01] J. Weickert, J. Heers, C. Schnörr, K. J. Zuiderveld, O. Scherzer, and H. S. Stiehl. Fast parallel algorithms for a broad class of nonlinear variational diffusion approaches. *Real-Time Imaging*, 7(1):31–45, February 2001.

[WII99] J. Weickert, S. Ishikawa, and A. Imiya. Linear scale-space has first been proposed in Japan. *Journal of Mathematical Imaging and Vision*, 10(3):237–252, May 1999.

[Win03] G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods.* Springer, Berlin, 2003.

[Wit83] A. P. Witkin. Scale-space filtering. In *Proc. Eighth International Joint Conference on Artificial Intelligence*, volume 2, pages 945–951, Karlsruhe, West Germany, August 1983.

[WS01a] J. Weickert and C. Schnörr. A theoretical framework for convex regularizers in PDE-based computation of image motion. *International Journal of Computer Vision*, 45(3):245–264, December 2001.

[WS01b] J. Weickert and C. Schnörr. Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14(3):245–255, May 2001.

[WSM$^+$05] J. Weickert, G. Steidl, P. Mrázek, M. Welk, and T. Brox. Diffusion filters and wavelets: what can they learn from each other. In N. Paragios, Y. Chen, and O. Faugeras, editors, *Mathematical Models of Computer Vision: The Handbook.* 2005. To appear.

[WtHV98] J. Weickert, B. M. ter Haar Romeny, and M. A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, March 1998.

[WWS05] M. Welk, J. Weickert, and G. Steidl. A four-pixel scheme for singular differential equations. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale Space and PDE Methods in Computer Vision*, number 3459 in Lecture Notes in Computer Science, pages 610–621. Springer, Berlin, April 2005.

[WZG02] Y.-N. Wu, S.-C. Zhu, and C. Guo. Statistical modeling of texture sketch. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Proc. 7th European Conference on Computer Vision*, volume 2352 of *Lecture Notes in Computer Science*, pages 240–254, Copenhagen, Denmark, May 2002. Springer.

[YD99] Y. Yacoob and L. S. Davis. Temporal multi-scale models for flow and acceleration. *International Journal of Computer Vision*, 32(2):1–17, September 1999.

[YG88a] A. L. Yuille and N. M. Grzywacz. A computational theory for the perception of coherent visual motion. *Nature*, 333:71–74, 1988.

[YG88b] A. L. Yuille and N. M. Grzywacz. The motion coherence theory. In *Proc. Second International Conference on Computer Vision*, pages 344–354, Washington, DC, December 1988. IEEE Computer Society Press.

[YK96] Y.-L. You and M. Kaveh. Anisotropic blind image restoration. In *Proc. 1996 IEEE International Conference on Image Processing*, volume 2, pages 461–464, Lausanne, Switzerland, September 1996.

[You71] D. M. Young. *Iterative Solution of Large Linear Systems*. Academic Press, New York, 1971.

[YS01] A. Yezzi and S. Soatto. Stereoscopic segmentation. In *Proc. 8th International Conference on Computer Vision*, volume 1, pages 59–66, Vancouver, Canada, July 2001.

[Zal79] S. T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, 31:335–362, 1979.

[ZCMO96] H. K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127:179–195, 1996.

[ZGWW02] S.-C. Zhu, C. Guo, Y. Wu, and W. Wang. What are textons? In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Proc. 7th European Conference on Computer Vision*, volume 2353 of *Lecture Notes in Computer Science*, pages 793–807, Copenhagen, Denmark, May 2002. Springer.

[ZY96] S.-C. Zhu and A. Yuille. Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900, September 1996.