INAUGURAL-DISSERTATION

zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich–Mathematischen Gesamtfakultät
der
Ruprecht–Karls–Universität
Heidelberg

vorgelegt von
Annette Stahl
aus der Sickingenstadt Landstuhl

Tag der mündlichen Prüfung: 12. Oktober 2009

Dynamic Variational Motion Estimation and Video Inpainting with Physical Priors

Gutachter:	Prof. Dr. Christoph Schnörr
Zweitgutachter:	Prof. Dr. Bernd Jähne

To Christian,
my parents Elisabeth and Konrad,
my brothers Ludwig and Jochen

Abstract

This thesis belongs to the field of image sequence analysis in computer vision with the main focus to propose and to explore new motion estimation approaches. The main contribution is the incorporation of the Burgers regularisation term for variational motion estimation approaches exploiting physical prior knowledge that is new in the field of image sequence processing. Using one of the motion estimation approaches along with an appropriate transport process we also propose a new reconstruction approach for missing data in image sequences, also known as video inpainting.

We exploit and extend the existing framework of standard variational optical flow approaches, which we use to recover optical flow fields from image sequences by minimising an appropriate energy functional. In general, these energy functionals consist of two terms: a *data term* that imposes e.g. for optical flow approaches the brightness constancy assumption and a *regularisation term* that enables us to incorporate prior knowledge into the energy functional. We concentrate on the latter term for the exploration of new priors based on physical transport processes.

In particular we present a novel partial differential equation based representation for image motion computation. The Burgers equation is employed in order to obtain a physical plausible regularisation term for dynamic image motion modelling. The resulting distributed-parameter approach incorporates a spatio-temporal regularisation in a recursive online fashion, contrary to previous variational approaches which are designed to evaluate the entire spatio-temporal image volumes in a batch processing mode. The inertia behaviour of the Burgers equation leads to a temporal filtering effect that is superior to common spatio-temporal approaches with respect to missing image measurements. We also speculate about relations of our approach to visual perceptual phenomena like the motion aftereffect.

Furthermore, we design an optimal control approach for image sequence processing using the two-dimensional Burgers equation as a constraint equation. The flow fields are forced towards vector fields which satisfy the Burgers equation. In order to estimate the motion of apparent velocities of image measurements in an image sequence, control variables are included and determined by minimising an appropriate objective functional. Control variables and optical flow adjust to the observed image data.

For computer vision applications these new approaches show an often desired property that might be exploited to design an attentional mechanism within a superordinate processing stage: Un-modeled and therefore unexpected motion events lead to an increased force field that is localised around the unexpected motion and may therefore help to identify an event of interest.

We exploit some of the above developed ideas to obtain a new algorithm for video inpainting to reconstruct damaged parts in image sequences. Image information is transported in a structure preserving way into the inpaint region using a fluid based advection-diffusion equation. The first results are very promising and motivate the development of more sophisticated video processing algorithms.

We experimentally explore the properties and the potential benefits of the presented variational approaches. The most prominent features that our approaches show are the inertia behaviour along with its temporal filtering effect and the composition of sharp motion boundaries in motion direction which is an often desired property in optical flow computations.

Zusammenfassung

Diese Arbeit ist einzureihen in das Wissenschaftsgebiet der Bildverarbeitung, das sich unter anderem mit der Analyse von Bildfolgen befaßt. Wir sind daran interessiert neue Ansätze zur Bewegungsschätzung in Bildfolgen vorzustellen und zu untersuchen. Der grundlegende Beitrag dieser Arbeit ist die Einführung des Burgers-Regularisierungsterms, der im Rahmen der variationellen Bewegungsschätzung von Bildfolgen physikalisches Vorwissen für die Bewegungsschätzung beisteuert, das in dieser Art und Weise neu im Bereich der Bildfolgenverarbeitung ist. Unter Verwendung dieser Ansätze und unter Einbindung einer anderen physikalischen Transportgleichung stellen wir auch einen neuen Ansatz zur Rekonstruktion von fehlenden Bilddaten in Bildfolgen vor.

Wir benutzen und erweitern das bestehende Rahmenwerk von herkömmlichen Variationsansätzen um das optische Flussfeld mittels der Minimierung eines geeigneten Energiefunktionals zu erhalten. Diese Energiefunktionale bestehen aus zwei Haupttermen: Dem Datenterm, der für optische Flussansätze im Allgemeinen auf der Annahme der Grauwerterhaltung basiert, und dem Regularisierungsterm, der uns die Möglichkeit bietet Vorwissen, basierend auf physikalischen Transportgleichungen in die Energieminimierung mit einzubinden, im speziellen die Burgersgleichung, welche eine grundlegende Gleichung in der Strömungsmechanik darstellt. Die physikalischen Eigenschaften der Gleichung repräsentieren Gleichförmigkeit und Kontinuität der Bewegung, sowie das Trägkeitsverhalten bewegter Objekte in Bezug auf Bewegungsänderung.

Speziell präsentieren wir eine neue auf einer partiellen Differentialgleichung basierende Darstellung zur Berechnung von Bewegung in Bildfolgen. Darin wird die Burgersgleichung als physikalisch plausibler Regularisierungsterm zur Modellierung dynamischer Bewegung in Bildern eingesetzt. Im Gegensatz zu herkömmlichen Raum-Zeitlich regulierten Anzätzen, wo das ganze Bildvolumen zur Bewegungsschätzung vorliegen muss, werden die Bewegungsfelder in

dem resultierenden Ansatz rekursiv und online berechnet. Die charakteristische Trägheitseigenschaft der Burgersgleichung führt zu einem zeitlichen Filterungseffekt, welcher dem Ansatz bei fehlenden Bilddaten zu Gute kommt und Bewegungsschätzungen zuläßt auch wenn Daten fehlen. Im Weiteren weisen wir auf die Anwendbarkeit unseres Ansatzes im Bereich der visuellen Wahrnehmung hin, im speziellen der Modellierung des Phänomens der Nachwirkungserscheinungen von Bewegungen.

Darüber hinaus, formulieren wir einen Kontrollansatz zur Bewegunsschätzung, in welchem die Vektorfelder ebenfalls der zweidimensionalen Burgersgleichung genügen sollen. Dazu führen wir Kontrollvariablen ein, die den optischen Fluss so kontrollieren, dass er der Bewegung der gemessenen Bilddaten folgt und möglichst der Burgersgleichung genügt. Um die Bewegung in den Bilddaten zu schätzen wird eine geeignete Zielfunktion bezüglich der Bewegungsvariablen und Kontrollvariablen minimiert.

Die neuen Ansätze zeigen eine Eigenschaft die für Anwendungen in der Bildverarbeitung ausgenutzt werden kann: Nicht vom Modell vorhergesehene und daher nicht erwartete Bewegungen können einem übergeordneten Kontrollmechanismus mitteilen, dass eine ungewöhnliche Bewegung vorliegt und eine entsprechende Aktion eingeleitet werden kann.

Basierend auf den Grundlagen dieser neuen Ansätze entwickeln wir einen neuartigen Algorithmus zur Rekonstruktion beschädigter Bildregionen in Bildfolgen. Die fehlende Bildinformation wird strukturerhaltend aus den umliegenden nicht beschädigten Regionen des Bildes mit Hilfe der Bewegungsschätzung und einem weiteren physikalischen Transportmechanismus in die Fehlstellen transportiert. Die ersten Ergebnisse dieses Ansatzes sind vielversprechend und können die Basis für fortgeschrittene Anwendungen bilden.

Mit unseren experimentellen Untersuchungen der resultierenden Ansätze haben wir die neuen Eigenschaften und ihre potenziellen Nutzen abgeschätzt. Die wohl wichtigste Eigenschaft basiert auf der Transport- und Trägheitseigenschaft der Burgersgleichung und dem damit einhergehenden zeitlichen Filterungseffekt. Aber ebenso die Herausbildung klarer Bewegungsgrenzen in Bewegungsrichtung stellt eine oft erwünschte Eigenschaft für optische Flussansätze dar.

Acknowledgements

First of all, I would like to thank Prof. Dr. Christoph Schnörr for providing me the opportunity to work at the Computer Vision, Graphics and Pattern Recognition (CVGPR) group (now Image & Pattern Analysis Group) and for supervising my dissertation in the interesting world of computer vision. The discussions provided me with deep insight into many different fields of mathematical methods, computer vision techniques and psychological understanding of vision. *Thanks a lot!*

I am also grateful to Prof. Dr. Bernd Jähne for serving as the second reviewer of my thesis.

Especially, I would like to thank the system administrators of the CVGPR group, Christian Gosch and Christian Schellewald, who spend a lot of time – removed from their research time – to do work that is often underestimated and rarely appreciated.

I would like to thank all members of the CVGPR group for creating a pleasant working atmosphere. In particular, I am very thankful to Paul Ruhnau and Christian Schellewald who read preliminary versions of this thesis and provided very helpful comments to improve my thesis. I would like to thank Paul Ruhnau, Christian Schellewald, Thomas Schüle and Christian Gosch for very inspiring discussions. I specially value the friendship of many of my colleagues: Daniel Cremers, Christian Gosch, Jens Keuchel, Timo Kohlberger, Paul Ruhnau, Christian Schellewald, Thomas Schüle, Stefan Weber, and Jing Yuan. Several trips and events with them are unforgettable.

Furthermore, I would like to thank my very good friend Marianne Steffen to being there in good and bad times, and Hans Wismath and Michael Mayer for their long time friendship.

Finally, I am very grateful to my loving family, my parents Elisabeth and Konrad Stahl, my brothers Ludwig and Jochen, with their wonderful families: Myriam, Heike, Karolin and Lena. And my aunts and uncles, Maria and Ludwig Goldinger with Paul and Felix, and Anni and Heinz Schwamberger with Sabine and Amay. All of them provided me with the support and love I needed. And thanks a lot to the open-hearted Schellewald family: Ulla, Armin, Andreas, Kerstin, Martin, Conny, Alioscha, Josef, Anneliese and Hildegard.

Especially, I want to thank Christian Schellewald, with whom it is wonderful to share the valuable time of life within this beautiful universe. Thank you for your infinite support, great charisma and all your love.

Contents

1

Introduction

1.1 Motivation

In this thesis we are concerned with the estimation of motion in image sequences, which poses one of the fundamental problems in computer vision. *Motion estimation* refers to the technique to compute a motion vector field between subsequent image frames in a video sequence. We intend to explore new ideas that incorporate *physical prior* knowledge about "motion" into a *variational* framework.

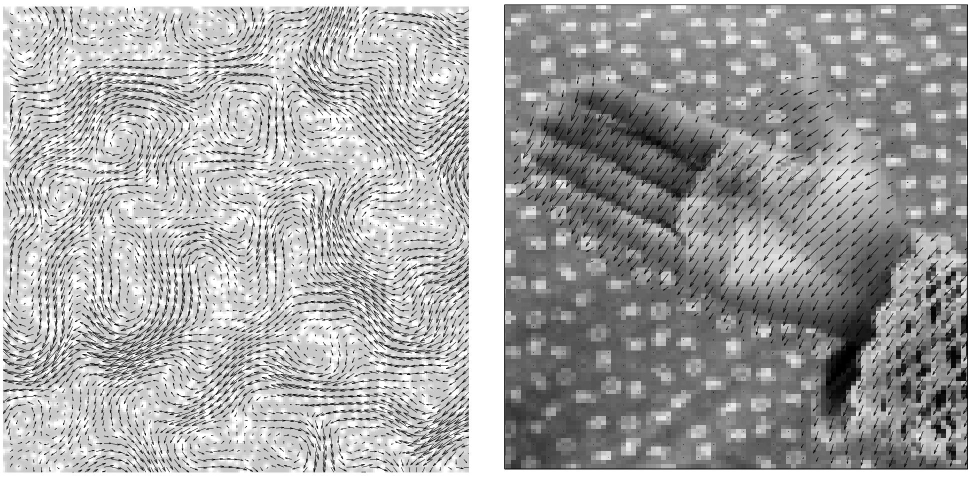


Figure 1.1: Motivation: We intend to transfer ideas from fluid flow estimation based techniques to the field of image sequence processing for every day image scenes. **Left:** Recovered velocity field for a typical Particle Image Velocimetry (PIV) image frame (background) using the variational flow estimation approach [117] with physical prior knowledge. **Right:** Optical flow estimation for a moving hand using physical prior knowledge (3.4).

Our motivation is based on fluid flow estimation techniques as proposed in [118] where we were concerned with extracting highly non-rigid velocity fields from image sequences of fluids (see left image of fig. 1.1).

Successful approaches that were originally developed in the field of computer vision within in the last decade to compute such flow fields are standard variational optical flow approaches. Higher-order regularisation techniques are used to improve the accuracy of such approaches. However, most of these approaches do not incorporate physical prior knowledge about the fluid flow itself. A physically plausible variational approach has to consider the image measurements - which are retrieved from the image data - along with an appropriate state equation - which models the physically behaviour of the fluid motion. As a consequence, we want to find a velocity field that is consistent with the observed image data and which also satisfies the physical model equation.

The main aim of this thesis is to transfer physically motivated fluid flow estimation techniques to the field of image sequence processing for every day image scenes. Therefore, we have to employ a state equation which is suitable to model also non-fluid motion, like traffic scenes or gestures (moving hand in fig. 1.1 right). The obvious question is, which model equation is appropriate for this kind of motion? - Considering fluids, we know from physics that the governing equations are given by the Navier-Stokes equations. However, it becomes clear, that the incompressible Navier-Stokes equations are not the best choice to model every day image scenes. For example, every day image scenes seem to be rigid in comparison with liquids like water or honey. Furthermore such scenes cannot be considered to be divergence free as a divergent optical flow field occurring in zoom-sequences easily reveals.

However, in this work we consider the scene as a ”fictive fluid” - assuming that its motion can be described by an appropriate model equation. The basic knowledge about image motion we intend to incorporate is that ”structures do not jump!”. Considering a rigid constant moving object one can determine that structures are transported by a velocity field and along with it the velocity field is transported by itself. The physical model equation, which exactly describes this behaviour is known as *Burgers equation* and allows to model the movement of more rigid objects. The Burgers equation provides us with a novel PDE-based representation of image motion exploiting this knowledge and allows us to present a *dynamic* viewpoint of image motion processing. The incorporation of this new prior results in a variational approach for dynamic image motion with physical prior, where the observed image measurements force the fictive fluid into the appropriate motion direction. In particular that means that the approach favours a velocity field that is in accordance with the observed image data and additionally satisfies the Burgers equation.

The properties of our new physical prior encouraged us to exploit them for other applications in image sequence processing as well. In computer vision the aim of inpainting is to ”reconstruct” image information of image parts - where information is damaged or even completely missing - in a visually undetectable manner. In this thesis we explore our model equation in combination with *video inpainting*.

In history, several different media types have been used to record and store motion pictures. The most prominent example is probably the flexible celluloid for motion pictures and movies but digital storage solutions are likely to dominate in the future. Watching in particular older films one can often see faults like flickering black or white patches on the celluloid frames. These patches are sometimes just markers but originate also from a degeneration of the material. In extreme cases, even several frames can be lost completely due to a projector fault which damaged that part of the film material. In connection with this and in arts - and less pleasantly in censorship - image and video inpainting has been used to retouch images and videos to remove objects. When the film data is available in digital form, computer implemented inpainting methods are desired to repair the damaged image sequence. In this thesis we explore a new idea to use physically motivated priors within motion estimation techniques to transport image measurements in regions of missing image data. We make use of two model equations, the previous mentioned Burgers equation and the advection-diffusion equation. Due to the properties of the Burgers equation it is - to a high degree - possible to transport the motion field in regions with missing image measurements within the sequence. The resulting velocity field is used in connection with the advection-diffusion equation to transport available image measurements in a structure preserving way into the inpaint region. The approach and early evaluation results are presented.

We would like to emphasise, that the present work is an attempt to innovatively adopt established methods from the field of applied mathematics in a new form to image sequence analysis and processing in order to evaluate the capability of the resulting approaches. Therefore, the presented applications to image motion estimation are novel and explorative.

1.2 Related Work

1.2.1 Particle Image Velocimetry

The root of our research can be found in methods that were originally developed in the field of computer vision and that were modified for the purpose of Particle Image Velocimetry (PIV) applications. Although PIV is not directly within the scope of this thesis one of our contributions lies in this field and we refer to some work in this research field.

Particle Image Velocimetry is an optical method to measure velocities or related quantities in fluids. So called tracer particles are added to the fluid and high-speed cameras are used to capture slow motion fluid image sequences. Velocity fields are measured by determining the displacement the individual particles have been made within this time. Several methods were developed to analyse these kind of particle image pairs and sequences. For a survey on PIV the reader is refered to [111]. We focus on computer vision methods that are related to the topics in this thesis. These are in particular methods that are based on variational optical flow estimation techniques, where the displacement field of corresponding particles in subsequent frames of an image sequence are known as *optical flow* (cf. sec. 2.1). The flow estimation is based on the assumption that the image intensities are preserved between consecutive image frames. The variational framework allows to include physical prior knowledge in a natural way. The PhD-thesis of Paul Ruhnau [115] is largely concerned with PIV methods using computer vision techniques and we wish to refer the reader interested in this field to his work.

1.2.2 Motion Estimation

Our main research interest is concerned with the estimation of motion in image sequences. This represents a vast research field in computer vision, but we refer only to [74, 3], which provide nice introductions into this field. An overview of some methods that are related or relevant for our work is given below.

Local Optical Flow Methods are designed to compute the optical flow at a certain pixel position by using only the image information in the local neighbourhood of this specific pixel. The first local approach was introduced by Lucas and Kanade [88] in 1981. They made the assumption that the optical flow is constant within a small window around the central pixel location at any time. Due to their locality the computation of such approaches is usually quite efficient. However, in this work we are mainly interested in variational optical flow methods which are intended to find a global optimum. Nevertheless, in section 2.3 we will describe local optical flow approaches. A review of some different types of local optical flow approaches along with their extensions can be found in [74].

Variational Optical Flow Methods represent global optimisation problems which are used to recover the optical flow field from an image sequence as a minimiser of an appropriate energy functional. In general, these energy functionals consist of two terms: a *data term* that imposes, e.g. a brightness constancy assumption and a *regularisation term* which impose additional constraints like global or piecewise smoothness to the optical flow field.

One of the first variational methods for motion analysis was introduced by Horn and Schunck [67] in 1981 and incorporates a *homogeneous* regularisation term, where the optical flow is enforced to vary smoothly in space. As the name suggests the smoothing is done in a homogeneous way, which leads to an undesired blurring across flow discontinuities. Therefore, other smoothness terms were introduced to regularise the flow in an *image-driven* [99, 119, 2] or *flow-driven* [120, 40, 142] way, where the flow is suppressed to smooth across object boundaries or motion boundaries, respectively. Furthermore, the employed diffusion processes can be divided into *isotropic* and *anisotropic* diffusion approaches. A systematic classification of these approaches can be found in [142], where the authors also suggest the following classification. According to this, image-driven approaches belong to the class of *quadratic* models [12] and flow-driven approaches correspond to the class of *non-quadratic* regularisers [120]. Note that homogeneous and image-driven regularised methods (which we describe in chapter 2) are used in this work as foundation to incorporate physical prior knowledge to regularise image motion in a spatio-temporal way. Further related work can be found in chapter 3.

Most of the variational approaches enforce a *spatial* regularisation of the flow. However, some efforts have been made to incorporate *spatio-temporal* smoothness [101] by enlarging the integration domain of the variational approach into the temporal direction. The work of [143] investigates an extension of spatial flow-driven regularisation terms to spatio-temporal flow-driven regularisers. In these approaches, time is considered as a third dimension analogue to the two spatial dimensions. This lead to a three-dimensional elliptic PDE with a static description of motion, where a flow vector is determined by the solution of a linear system involving the data of the full 3D image volume. Solving this variational problem allows the computation of the optical flow only after the last frame of the image volume has been captured. However, these kind of approaches improve both the robustness and the accuracy of the motion estimation.

In chapter 4 we present an optimal control based formulation for the optical flow computation subject to a physical constraint. Optical flow fields are forced to satisfy a specific model equation. The model equation includes control variables that allow to adjust the image motion in such a way that it fits to the apparent velocities of the moving objects in the given image sequence. Our motivation draws on literature on the control of distributed parameter systems in connection with fluid dynamics [58] and our intention is to apply these methods to image motion processing [116, 16]. This represents a new approach in computer vision and for work related to that area we refer to section 4.1.2.

1.2.3 Inpainting

Inpainting methods try to reconstruct missing data in digital images or sequences. The applications range from removing objects in images to reconstructing damaged parts in images and photographs. In the field of computer vision many methods were proposed to reconstruct image data in images or video sequences. Here our focus lies on PDE-based image and video inpainting methods:

Image Inpainting Methods The term *digital inpainting* was first introduced to image processing by Bertalmío, Sapiro, Caselles and Ballester [10] and has its origin in the restoration of artwork from the Renaissance. The idea is to propagate information from the surrounding area continuously into the region of missing image data (inpaint region). To achieve this, different strategies can be found in literature. Total Variation (TV) methods [27, 30] which minimise the total variation norm, are used to reconstruct damaged parts in images in a smooth but edge preserving way. In [10, 89] the authors propose techniques which are based on the propagation of level curves into the inpaint region. These methods and the work of [9] is related to our approach (which we will describe in chapter 6) as both exploit a physically transport equation to transport image information into the inpaint region. However, our approach employs the transportation not only in a spatial context but also in a temporal. In chapter 5 we are concerned with some particular image inpainting algorithms.

Video Inpainting Methods In video inpainting the aim is to restore missing data in video or image sequences. Image inpainting is used as a fundamental. One of the first video inpainting approaches was proposed by Bertalmío et al. [9] but is essentially an image inpainting approach that is applied frame-by-frame for the entire sequence. Therefore, it belongs to the category of purely *spatial* approaches. Temporal extensions were proposed to incorporate information that the image sequence provides over time. Different techniques have been used to extract temporal information of the image sequences. The authors of [123] proposed a method where the colour or intensity information from previous frames of the sequences are just copied directly into the inpaint region of the following frames. In [144] the authors suggested a method where missing image values are constrained to form coherent structures with respect to reference samples. Instead of transferring intensity measures directly the approaches presented in [56, 124] extract the motion information of the image sequence in order to transfer the motion field into missing areas. The motion field is recovered by using local or global optical flow methods. The spatio-temporal video-inpainting approach, we present in chapter 6 uses a variational optical flow technique to recover the velocity field of the image sequence, which is then used within a physically transport equation to transport the surrounding image information into the region of missing image data in a spatio-temporal manner. Other methods are discussed in chapter 6.

1.3 Contributions

Our main research interest in this work is to develop new techniques for variational motion estimation. The idea is to transfer approaches known from fluid flow computation using physical prior knowledge to image sequence processing. Besides motion estimation itself these techniques are applied in a video inpainting approach. Below we describe our contributions to the appropriate fields in computer vision

Variational Motion Estimation Our starting point are standard variational optical flow approaches as presented in chapter 2 which are mainly spatial regularised. We adapt in chapter 3 the spatial regularised approach of Nagel (cf. sec. 2.2.3), which extends the basic approach of Horn and Schunck (cf. sec. 2.2.2), to an anisotropic image-driven diffusion process and present a novel PDE-based representation of image motion. Our approach performs spatio-temporal regularisation in a recursive online fashion, unlike previous variational approaches evaluating entire spatio-temporal image volumes in a batch processing mode (cf. 2.2.4).

Our major contribution in this thesis is to introduce the Burgers equation as a new regularisation term as physically plausible description for motion within a variational framework and to explore the capabilities and properties of the resulting approaches. In chapter 3 we present a variational approach to motion estimation for image sequence processing with the Burgers equation as new physical prior [126]. We are able to determine five new and for the field of computer vision relevant properties: (i) The compressible Burgers equation is employed in order to obtain a physical plausible regularisation term for dynamic image motion. This allows to change the static viewpoint of image motion processing to a dynamical one. (ii) Spatio-temporal regularisation is performed in a sliding window manner, which allows recursive online (non-batch) computation. (iii) The inertia behaviour of the Burgers equation leads to a temporal filtering effect that is superior to common spatio-temporal approaches. (iv) Deviations from the expected velocity distribution generate vector fields that may serve as attentional mechanism for a superordinate processing stage. (v) We briefly speculate about relations of our approach to perceptual phenomena like motion aftereffects.

Control based Motion Estimation In chapter 4 we design an optimal control approach for image sequence processing incorporating the two-dimensional Burgers equation as a constraint equation to obtain a dynamical motion evolution over time. In order to estimate the motion of apparent velocities of image measurements in an image sequence, control variables are included and determined by minimising an appropriate objective functional. The solutions are obtained by minimising the Lagrangian functional of the optimal control formulation.

We present this approach as previous mentioned as an attempt to adopt respective methods from the field of applied mathematics in a new form to image sequence processing to evaluate the capability of these kind of approaches.

Video Inpainting In the last two decades inpainting applications became popular as they provide new tools for digital imaging software, allowing to edit images in an easy way. Damaged images or videos can be easily reconstructed or even objects can be removed with only a few manual interaction.
The spatio-temporal video-inpainting approach, we present in chapter 6 uses ideas from PDE-based inpainting methods. To minimise the user interaction we designed a new two-step algorithm, where in the first step an optical flow field is recovered between subsequent image pairs and in the second step the velocity field is used within an advection-diffusion equation to transport image measurements into the region of missing image data. The key advantage of our new video-inpainting approach is the spatio-temporal regularisation where image

information from previous frames can be transported into the inpaint regions preserving the structural information. The incorporated physically transport equations are used to transport the optical flow and the image data into the inpaint region over time.

Particle Image Velocimetry Our motivation is based on a variational approach to motion estimation of instationary fluid flows for PIV applications. The approach employed the full incompressible Navier-Stokes equation as plausible physical prior knowledge. In connection with this work we presented in [118, 117] a variational approach for motion estimation of instationary fluid flows from image sequences. My contribution to that work lies in the incorporation of the vorticity transport equation that I implemented to explore an inpainting approach for a reconstruction of medical images. The vorticity transport equation is used within this framework to approximate two dimensional vector fields of turbulent curling fluids. The optimisation is done in a receding horizon manner contrary to previous approaches using spatio-temporal regularisation. This results in a recursive on-line implementation.

1.4 Organisation

This thesis is divided into three main parts:

In the first part I we are concerned with variational motion estimation techniques and approaches. Therefore, we present in chapter 2 the fundamental framework for motion estimation and present standard variational approaches. Our new distributed parameter approach for dynamic image motion is described and evaluated in detail in chapter 3. We introduce a new temporal regularisation framework, based on the Burgers equation as physical prior knowledge. A control based optical flow approach is introduced in chapter 4 subject to the same physically constraint equation.

The second part II of this thesis has image and video inpainting as main topics. In chapter 5 we are concerned with the general challenges of image inpainting and present some well-known approaches. A classification of image inpainting approaches which is based on the technique underlying the individual approaches can be found in chapter 5.1.2. This chapter serves mainly as a foundation for our video inpainting approach proposed in chapter 6. There we present an inpainting approach which is based on the physical Burgers transport mechanism and exploits the distributed parameter approach for image motion which incorporates dynamical prior knowledge. This results in a spatio-temporal video inpainting approach that can be implemented by a two step algorithm applied on a sliding window.

The third part III of this thesis includes the numerical background, solution and implementation of the approaches that are essential for this thesis. An overview of the fundamental model equations we exploit in out research is given in chapter 7 while chapter 8 discusses appropriate discretisation techniques for these equations. Chapter 9 provides an overview of appropriate numerical solvers that are used to solve the discretised problems.

We conclude this thesis with chapter 10 by summarising our work and discussing open problems. Furthermore we propose possible extensions for further work in our research area. The appendix contains supplementary material for chapter 4.

Part I

Variational Motion Estimation

2

Standard Variational Approaches

This chapter is organised as follows: After explaining what is considered as optical flow in this thesis we define the optical flow constraint equation. We introduce variational approaches in general and present the standard variational approach of Horn and Schunck with exemplary numerical solution. Furthermore, we present other approaches, which provides us with regularisation techniques that are used within this thesis.

2.1 Preliminaries

Motion analysis in computer vision deals with the analysis and estimation of the visual motion that occurs in image sequences or videos like the one shown in figure 2.1.

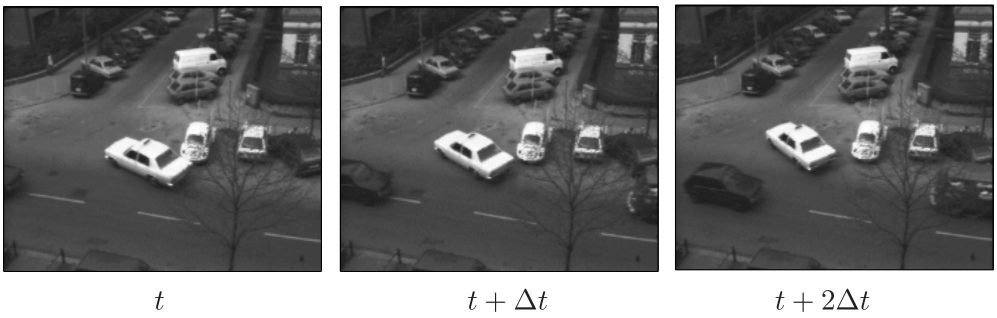

t $t + \Delta t$ $t + 2\Delta t$

Figure 2.1: Real image sequence: The well-known "Hamburg taxi scene" consists of image frames showing a taxi coming from the right side which turns right into a street. Additionally, two other dark coloured cars enter the scene, one from the left and one from the right. Only three frames of the full sequence are depicted here.

In this traffic scene we easily observe the displacements of objects between consecutive images and interpret this as the "motion" of the objects. But one has to be aware that this *apparent motion* can be different from the true *motion field* one is usually interested in. The true motion field is the projection of the exact 3D velocity field of the scene to the image plane and we cannot measure it directly. An extreme case where the real motion is quite different from the apparent motion we perceive is depicted in figure 2.2.

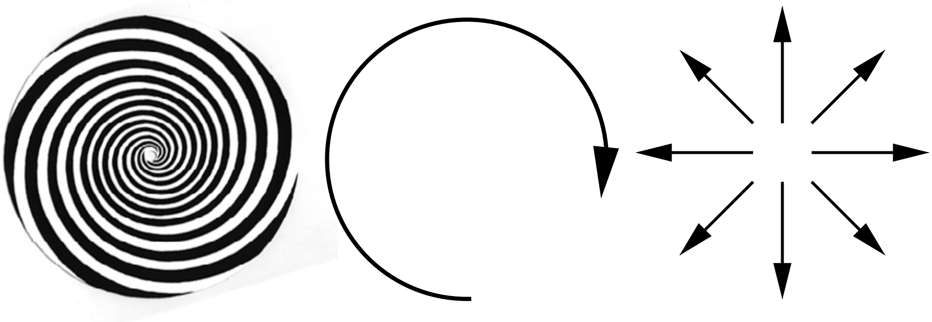


Figure 2.2: The real motion can be quite different from the apparent motion. **Left:** One frame of a "spiral" sequence which rotates clockwise. **Middle:** Indication of the true rotating 2D motion field. **Right:** The apparent motion we perceive is outwards.

If the pattern shown on the left hand side of figure 2.2 rotates clockwise we perceive an outward movement instead of the real rotational movement. Luckily, in common image sequences the apparent motion is a good approximation of the motion field. But nevertheless, one should keep in mind that the true motion field can only be approximated based on the given image data.

In computer vision the measurement of the apparent motion is known as *optical flow* or *image flow* and can be defined as follows:

Image flow is the velocity field in the image plane due to the motion of the observer, the motion of objects in the scene, or apparent motion which is a change in the image intensity between frames that mimics object or observer motion [75].

Several other similar definitions can be found in the literature [52, 68, 127]. Making use of the definition above we consider in this work the optical flow as a 2D velocity field that describes the intensity changes between images.

In the following we are concerned with techniques which have been used for the computation of the optical flow field between consecutive image frames of sequences. We focus on gradient based methods, which use spatio-temporal partial derivatives to estimate the optical flow at each point in the image and allow the derivation of variational methods.

2.1.1 Optical Flow Constraint

In this thesis we represent an image sequence by a real valued image intensity function $I(x,t)$ that is continuous in space and time. The variable $x = (x_1, x_2)$ denotes the location within a rectangular image domain Ω and $t \in [0,T]$ labels the corresponding frame at time t.

The starting point for most of the motion estimation approaches is the creation of a link between motion and the corresponding intensity variation. And the most frequent assumption within these approaches is that the observed intensity I is conserved over time. This means that the intensity at position (x_1, x_2) at time t will be the same as the intensity at time $t + \Delta t$ at position $(x_1 + \Delta x_1, x_2 + \Delta x_2)$ for a small Δt. Using the intensity function $I(x_1, x_2, t)$ along with $u_1(x_1, x_2)$ and $u_2(x_1, x_2)$, which are the two components of the optical flow vector $u = (u_1, u_2)^\top$ this results in the equation

$$I(x_1 + u_1\Delta t, x_2 + u_2\Delta t, t + \Delta t) = I(x_1, x_2, t), \tag{2.1}$$

where $\Delta x_1 = u_1\Delta t$ and $\Delta x_2 = u_2\Delta t$.

If we assume that the brightness varies smoothly over time the term on the left hand side of equation (2.1) can be approximated by a first-order Taylor expansion at the point (x_1, x_2, t)

$$\begin{aligned} I(x_1, x_2, t) + \Delta t\; u_1\; \partial_{x_1} I + \Delta t\; u_2\; \partial_{x_2} I + \Delta t\; \partial_t I \\ + \mathcal{O}(\Delta x_1^2, \Delta x_2^2, \Delta t^2) = I(x_1, x_2, t). \end{aligned}$$

Dividing by Δt and dropping the higher-order terms $\mathcal{O}(\Delta x_1^2, \Delta x_2^2, \Delta t^2)$ for $\Delta t \to 0$, we obtain the linearised constraint

$$\partial_t I + u \cdot \nabla I = 0. \tag{2.2}$$

This differential equation is well-known in the literature as the *Brightness Change Constraint Equation (BCCE)* or *Optical Flow Constraint Equation (OFCE)* and has been introduced by Horn and Schunck [67]. As the name suggests (2.2) represents a constraint on the optical flow components u_1 and u_2 which requires that the intensity of an object point stays constant along its motion trajectory. Consequently, this equation is an useful approximation for image sequences where the displacements are small, the grey value at a certain position is not influenced by global illumination changes, and where no occlusion occurs in the image sequence.

In order to deal with larger displacements one can use a hierarchical multiresolution approach which computes the optical flow – using a coarse to fine strategy – at each resolution level. We refer for more details to section 9.2 and 9.3.

In this thesis we do not focus explicitly on brightness change issues, but in order to cope with the violation of the brightness constancy assumption several approaches were proposed in the literature. Haussecker and Fleet [62] introduce a time-dependent physical process to estimate brightness variations in image sequences. In particular, they choose the heat transport equation as model equation to simulate the brightness variation as a diffusion process.

Negahdaripour and Yu [103] relax the brightness consistency assumption incorporating a brightness change model within their optical flow computation. They describe the image brightness at one point as a product of an illumination component and a surface reflectance term. An additional offset term compensates for irradiance, shading variations as well as for saturation of the sensor in dark or bright image regions. Black et al. [15] apply a probabilistic approach to model the changes in image appearance over time. The authors classify four generative models of appearance changes in connection with illumination variations, specular reflections and changes which might results from occlusion events or material properties of objects.

2.1.2 Aperture Problem

Another problem we have to deal with is the non-uniqueness of the optical flow constraint equation (2.2). This can easily be seen as the optical flow constraint is represented by a single linear equation with two unknown variables u_1 and u_2 at each location (x, t).

Writing the OFCE (2.2) as

$$u \cdot \nabla I = -\partial_t I, \tag{2.3}$$

we obtain that we are only able to determine the *normal flow* u_n (cf. fig. 2.3), which is the flow component in the gradient direction ∇I:

$$u_n = \frac{\nabla I}{|\nabla I|} \cdot u = -\frac{\partial_t I}{|\nabla I|}, \quad |\nabla I| > 0, \forall x.$$

Note that a direct estimation of the flow component perpendicular to u_n is not possible. This fact is known as *aperture problem*.

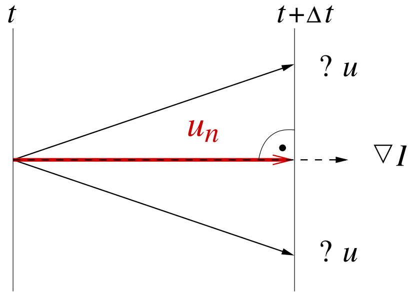


Figure 2.3: Only the normal flow u_n which represents the component of the optical flow in the direction of ∇I is determined by the optical flow constraint equation (2.2).

Figure 2.4 illustrates the aperture problem: The original sequence shows a black bar that moves from the upper-left corner to the lower-right one. Then the sequence is occluded by an image mask with a small window showing the original part of the scene. With that we are unable to determine the original movement of the bar anymore and only the motion in the direction of the gradient ∇I is perceived [68].

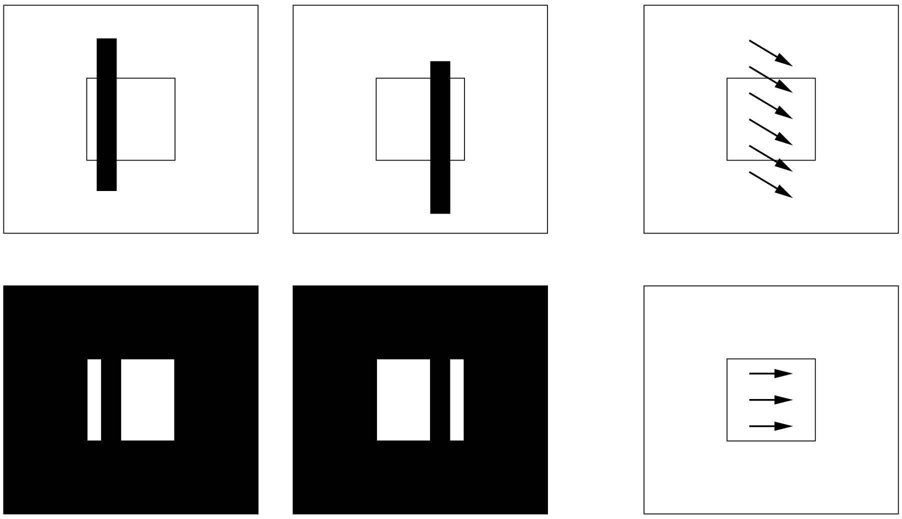


Figure 2.4: Illustration of the aperture problem. **First row:** Original movement of a bar. **Second row:** The same sequence is occluded with an image mask (black cover). The motion that is perpendicular to the grey value gradient ∇I cannot be perceived any longer. Only the components that are normal to the level lines of the image are determined.

According to Hadamard [60] a problem is *well-posed* if a solution exists, is unique and stable (the solution depends continuously on the initial data). Problems that are not well-posed in the sense of Hadamard are called *ill-posed*. It is clear that – due to the aperture problem – solving the optical flow constraint equation (2.2) belongs to the class of ill-posed problems.

Several ways to overcome the aperture problem and to transform the ill-posed optical flow problem into a well-posed one were proposed in the literature (see e.g [12, 3]). The authors of [12] state that in order to stabilise or regularise ill-posed optical flow formulations it is essential to introduce *additional constraints*. Approaches and regularisation techniques that are relevant for our work are presented in the following sections.

2.2 Variational Motion Estimation Approaches

Variational approaches turned out to be very useful for motion estimation problems in computer vision as they allow to introduce regularisation terms to the ill-posed optical flow constraint equation (2.2) in a clear mathematical way. In the following, we explain how standard variational approaches are usually applied for motion estimation.

2.2.1 Variational Formulation

A convenient method to solve problems in computer vision is to state and minimise an appropriate energy functional. For motion estimation problems this is done in a variational framework minimising an energy functional $E(u)$ with respect to u that consists of a data and a regularisation term:

$$\inf_u E(u) = \inf_u \left(E_{data}(u) + E_{regu}(u)\right). \tag{2.4}$$

For our motion related problems we make use of the following *data term*

$$E_{data}(u) = \int\limits_{\Omega} (\partial_t I + u \cdot \nabla I)^2 \, dx, \tag{2.5}$$

which comprises the optical flow constraint (2.2) and provides the link between the given image data and the desired velocity field. Note, as discussed previously, this problem is ill-posed as any vector field $u = (u_1, u_2)^\top$ satisfying (2.3) is a minimiser. Therefore a *regularisation term* $E_{regu}(u)$ is added to introduce additional constraints for the flow field u to obtain an unique solution.

In order to minimise (2.4), it is necessary that the first variation of the energy functional (2.4) must vanish

$$\frac{\partial E(u + \epsilon\tilde{u})}{\partial \epsilon}\Big|_{\epsilon=0} = \frac{\partial}{\partial \epsilon}\Big\{E_{data}(u + \epsilon\tilde{u}) + E_{regu}(u + \epsilon\tilde{u})\Big\}\Big|_{\epsilon=0} = 0. \tag{2.6}$$

If the energy functional $E(u)$ is convex this is a sufficient constraint for a global optimum.

The discretisation of this partial differential equation results typically in a sparse and positive definite system that can be solved using iterative solvers like the conjugate gradient method (see sec. 9.1), which belongs to the class of Krylov subspace methods.

The prototype (2.4) of a variational approach allows the use of various regularisation terms. In order to illustrate the numerical realisation of this approach – using a homogeneous smoothness term – we present below the pioneering work of Horn and Schunck. Details about variational methods for computer vision problems and regularisation techniques can be found in [121, 142, 3]. For particular variational approaches for image motion estimation we refer to [122] and references therein.

2.2.2 Global Approach from Horn and Schunck

Horn and Schunck [67] proposed their approach using the optical flow data term (2.5) and the regularisation term

$$E_{smooth}(u) = \int\limits_{\Omega} \alpha(|\nabla u_1|^2 + |\nabla u_2|^2) \, dx, \tag{2.7}$$

in order to regularise the optical flow constraint (2.2). This term incorporates prior knowledge about the optical flow field and enforces its smoothness *homogeneous*, therefore preferring neighbouring optical flow vectors to be similar. With this term – which was first introduced by Tikhonov and Arsenin in [132] – the full energy minimisation problem becomes:

$$E(u) = \frac{1}{2} \int\limits_{\Omega} \Big\{ (\partial_t I + u \cdot \nabla I)^2 + \alpha(|\nabla u_1|^2 + |\nabla u_2|^2) \Big\} dx. \tag{2.8}$$

The regularisation parameter $\alpha > 0$, $\alpha \in \mathbb{R}$ adjusts the relative importance of the smoothness term to the data term. With an increasing value of α the vector field is forced to become smoother. Note that functional (2.8) is strictly convex under weak assumptions and has an unique minimising vector field u [119]. The first variation of (2.8) yields

$$\begin{aligned} \frac{\partial E(u + \epsilon \tilde{u})}{\partial \epsilon}\Big|_{\epsilon=0} \quad = \quad & \int\limits_{\Omega} \Big\{ (\partial_t I + u \cdot \nabla I) \tilde{u} \cdot \nabla I \\ & + \alpha(\nabla u_1 \cdot \nabla \tilde{u}_1 + \nabla u_2 \cdot \nabla \tilde{u}_2) \Big\} dx = 0, \end{aligned} \tag{2.9}$$

which results in a constraint equation for the optimal optical flow field. Exploiting the Finite Element Method (FEM) the numerical solution can be obtained as summarised below. For more details we refer to section 8.5 and [34, 121].

Numerical Solution

We use piecewise linear finite elements to discretise the variational equation (2.9) and compute the global minimiser u by solving the associated system of linear elliptic equations.

The unique minimising flow field u is determined by the variational equation

$$a\big((u_1, u_2)^\top, (\tilde{u}_1, \tilde{u}_2)^\top\big) = b\big((\tilde{u}_1, \tilde{u}_2)^\top\big) \ , \quad \forall \tilde{u}_1, \tilde{u}_2 \ , \tag{2.10}$$

where

$$\begin{aligned} a\big((u_1, u_2)^\top, (\tilde{u}_1, \tilde{u}_2)^\top\big) = & \int\limits_{\Omega} \Big\{ \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \cdot \nabla I \nabla I \cdot \begin{pmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{pmatrix} \\ & + \alpha \Big(\nabla u_1 \cdot \nabla \tilde{u}_1 + \nabla u_2 \cdot \nabla \tilde{u}_2 \Big) \Big\} dx_1 dx_2 \ , \end{aligned} \tag{2.11}$$

and

$$b\big((\tilde{u}_1, \tilde{u}_2)^\top\big) = - \int\limits_{\Omega} \partial_t I \nabla I \cdot \begin{pmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{pmatrix} dx_1 dx_2 \ . \tag{2.12}$$

One of the simplest discretisation is obtained by choosing a regular triangulation of the image domain Ω and attaching to each pixel position a piecewise linear basis function $\phi(x_1, x_2)$ for each function $u_1, u_2, \tilde{u}_1, \tilde{u}_2$, as illustrated in figure 2.5.

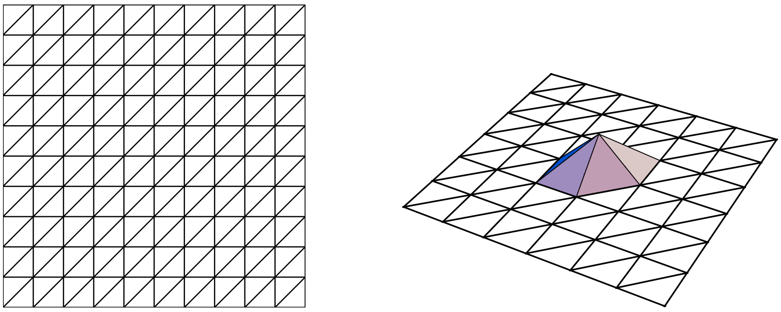


Figure 2.5: Left: Uniform triangulation of the image domain Ω. **Right:** Basis function $\phi_i(x_1, x_2)$ that belongs to a pixel position i.

Indexing each pixel position (k, l) by $1, 2, \ldots, N$ we thus have

$$u_1(x_1, x_2) = \sum_{i=1}^{N} u_i \phi_i(x_1, x_2) \ ,$$

and similarly for $u_2, \tilde{u}_1, \tilde{u}_2$. Hence, each of the functions $u_1, u_2, \tilde{u}_1, \tilde{u}_2$ is represented by N real variables. To simplify the notation, we use the same symbols to denote these vectors: $u_1, u_2, \tilde{u}_1, \tilde{u}_2 \in \mathbb{R}^N$. Inserting into (2.10) leads to

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \cdot A \begin{pmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{pmatrix} = b \cdot \begin{pmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{pmatrix} \ , \quad \forall \tilde{u}_1, \tilde{u}_2 \ ,$$

and hence we get the linear system

$$A \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = b. \tag{2.13}$$

The $2N \times 2N$–Matrix A factorises into

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{12}^\top & A_{22} \end{pmatrix} ,$$

where by virtue of (2.11)

$$\begin{aligned} (A_{11})_{k,l} &= a\big((\phi_k, 0)^\top, (\phi_l, 0)^\top\big) \\ (A_{12})_{k,l} &= a\big((\phi_k, 0)^\top, (0, \phi_l)^\top\big) \\ (A_{22})_{k,l} &= a\big((0, \phi_k)^\top, (0, \phi_l)^\top\big) \ . \end{aligned}$$

Analogously, the $2N$–vector b factorises into $b = (b_1^\top, b_2^\top)^\top$ where by virtue of (2.12)

$$\begin{aligned}(b_1)_k &= b\big((\phi_k, 0)^\top\big)\\ (b_2)_k &= b\big((0, \phi_k)^\top\big)\ .\end{aligned}$$

Note that the linear system (2.13) is sparse and positive definite (see also sec. 8.5.2). Thus u_1, u_2 can be conveniently computed by an appropriate iterative solver as explained in section 9.1.

Synthetically Example

We illustrate a computational result of the Horn and Schunck approach (2.8) for a synthetically image sequence. Figure 2.6 shows a “zoom in” office scene. The computed flow in the texture-less regions (e.g. the non-textured wallpaper) is reconstructed from the surrounding velocity information.

t $t + \Delta t$

Figure 2.6: Illustration of the Horn and Schunck approach: Computational result for an image pair of the synthetic ”Office - Cam Zooms in” sequence [93] which induces a divergent velocity field. **Left**: Image frame at time t. **Right**: Image frame at time $t + \Delta t$ along with the optical flow field u obtained for the image pair using the approach of Horn and Schunck (2.8), ($\alpha = 0.04$). Though a large portion of the image is non-textured, an accurate flow field can be computed at every location through the Horn and Schunck smoothness assumptions. Note that the office chair is closer to the camera - therefore its optical flow vectors are longer.

In regions with no texture information, the image gradient ∇I vanishes and no reliable motion field can be estimated from the image data. In such situations the smoothness term (2.7) dominates and leads to a flow field which obtains its information from neighbouring regions resulting in a *dense flow field*. The disadvantage of the Horn and Schunck regularisation term (2.7) is the homogeneous regularisation at *motion boundaries*. It leads to a smoothing over discontinuities - thus sharp contours are lost.

McCane et al. [93] mentioned in their work, where the performance of eight different optical flow algorithms are compared, Horn and Schunck as one of the best performers for motion estimation. However, currently the most accurate optical flow results are still obtained by variational optical flow approaches, for example approaches from Bruhn et al. [22], Brox et al. [21] and Papenberg et al. [107].

The smoothness assumption of Horn and Schunck is clearly violated at occlusion edges across which the optical flow field changes abruptly. As a real scene usually comprises several sharp motion boundaries the discontinuities of the flow field at motion boundaries is an important issue. The results can be improved by using anisotropic regularisation techniques like the one presented below. Other work on regularisation techniques, which preserve motion boundaries are [14, 35, 84, 104].

2.2.3 Image-Driven Regularisation

One example for an *image-driven* optical flow estimation method is given by the *anisotropic* approach of Nagel and Enkelmann [99, 102]. In order to regularise (2.2) they introduced the following directed smoothness term

$$E_{smooth}(u) = \int_{\Omega} \alpha \Big(\nabla u_1^\top D(\nabla I) \nabla u_1 + \nabla u_2^\top D(\nabla I) \nabla u_2 \Big) \, dx. \qquad (2.14)$$

Note that the regularisation term (2.7) of Horn and Schunck is a special case of (2.14), where $D(\nabla I)$ is just the unit matrix. Usually, $D(\nabla I)$ is a regularised projection matrix perpendicular to ∇I and is defined as

$$D(\nabla I) = \frac{1}{\|\nabla I\|^2 + 2\lambda^2} \Big(\nabla I^\perp (\nabla I^\perp)^\top + \lambda^2 \mathbf{1} \Big) \,, \qquad \mathbb{R} \ni \alpha, \lambda > 0.$$

Under weak assumptions the following functional

$$E(u) = \frac{1}{2} \int_{\Omega} \Big\{ (\partial_t I + u \cdot \nabla I)^2 + \alpha \Big(\nabla u_1^\top D(\nabla I) \nabla u_1 + \nabla u_2^\top D(\nabla I) \nabla u_2 \Big) \Big\} dx \quad (2.15)$$

is strictly convex and has an unique minimising vector field u [119]. The regularising term, introduced by Nagel, extends the basic isotropic approach of Horn and Schunck (2.8) to an anisotropic image-driven diffusion process (cf. [100, 122, 142]) which prevents the smoothing of motion boundaries at grey value edges. Note that the smoothing of the flow is essentially large in tangent direction of the isophotes when $|\nabla I| >> \lambda$. The smoothing behaviour depends mainly on the intensity and not on the motion itself.

We emphasise that the approaches from Nagel (2.15) and Horn and Schunck (2.8) takes only into account *spatial* context and determines a vector field for a *fixed* point in time $t \in [0, T]$. While it has been shown in [143] that a spatio-*temporal* extension of this class of approaches, using spatio-temporal gradients and a corresponding domain of integration $\int_{\Omega \times [0,T]} \dots dxdt$, improves both robustness and accuracy of the motion estimation. Below we describe that extension.

2.2.4 Spatio-Temporal Regularisation

Weickert and Schnörr [143] propose a modification of the two-dimensional Horn and Schunck approach (2.8) to three dimensions:

$$E(u) = \frac{1}{2} \int\limits_{\Omega \times [0,T]} \Big\{ (\partial_t I + (u \cdot \nabla) I)^2 + \alpha (|\nabla_\theta u_1|^2 + |\nabla_\theta u_2|^2) \Big\} dxdt, \qquad (2.16)$$

where ∇_θ is the spatio-temporal gradient $(\partial_{x_1}, \partial_{x_2}, \partial_t)^\top$ of the velocity components.

The complete flow field is determined by the solution of a linear system that involves the entire image volume. That means that the numerical realisation leads to a batch processing mode [1] where all variables of the entire spatio-temporal volume are determined by a large linear system of equations.

Figure 2.7 shows computational results for the well-known "Hamburg taxi scene" using approach (2.16).

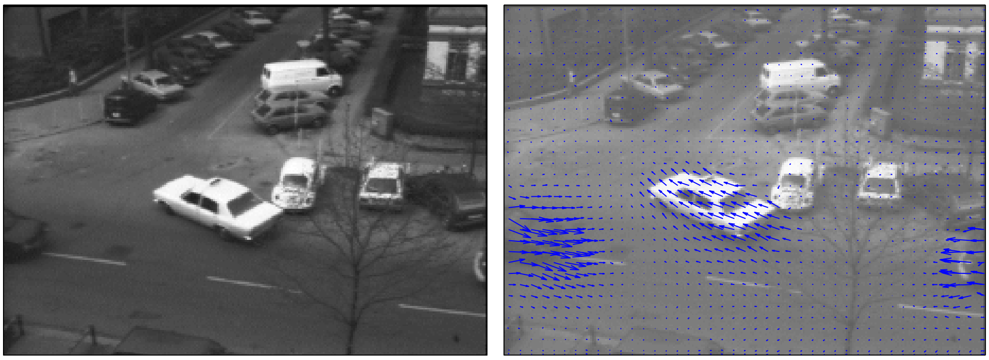
Beside the global approaches, where the optical flow problem is solved over the whole image domain, local approaches determine the flow at a specific pixel position $x = (x_1, x_2)^\top$ by using only the image information in the local neighbourhood. In order not to neglect these approaches the well-known approach from Lucas and Kanade is presented below.

2.3 Local Approach from Lucas and Kanade

The idea of a local approach is to compute the optical flow field at a certain pixel position x_0 using only a small defined neighbourhood $\mathcal{N}(x_0)$ of x_0. The local optical flow approach from Lucas and Kanade [88] exploits the assumption that the flow field $u(x) = u(x_0)$ is constant within a certain window and is formulated as minimisation of the following functional:

---

[1] The optical flow estimation depends at each processing step on the entire image sequence.



t $t + \Delta t$

Figure 2.7: Three-dimensional extension of the Horn and Schunck approach from Weickert and Schnörr (2.16). The image sequence consists of a subset with 14 frames (1, 4, 7,...,40) out of the original "Hamburg taxi scene". **Left**: Image frame 4. **Right**: Image frame 7 along with the computed optical flow field u using a coarse-to-fine approach (see sec. 9.2) to cope with the large motion. ($\alpha = 0.005$).

$$\begin{aligned} J(u) &= \sum_{x \in \mathcal{N}(x_0)} G_\rho(x - x_0) \left(\partial_t I(x) + u \cdot \nabla I(x)\right)^2 \\ &= \sum_{x \in \mathcal{N}(x_0)} G_\rho(x - x_0) \Big(\partial_t I^2(x) + 2\partial_t I(x) \nabla I(x)^\top u + u^\top \nabla I(x) \nabla I(x)^\top u\Big) , \end{aligned}$$

where G_ρ denotes a Gaussian distribution with variance ρ. Computing the derivatives with respect to u and setting them to zero, in order to minimise $J(u)$, leads to

$$G_\rho \star \begin{pmatrix} (\partial_{x_1} I)^2 & \partial_{x_1} I \partial_{x_2} I \\ \partial_{x_2} I \partial_{x_1} I & (\partial_{x_2} I)^2 \end{pmatrix} u = -G_\rho \star \begin{pmatrix} \partial_t I \partial_{x_1} I \\ \partial_t I \partial_{x_2} I \end{pmatrix} . \tag{2.17}$$

System (2.17) is not necessarily uniquely determined. There is no solution in image regions with homogeneous image information or at image edges. Note that the assumption $u(x) = u(x_0)$ for $x \in \mathcal{N}(x_0)$ is often violated due to rotations and/or motion along motion boundaries and discontinuities.

To exchange the brightness change constraint equation (2.2) enhanced brightness change models are introduced in the literature, we refer to [62] and the references therein. A lot of research on local optical flow methods are proposed, which lies, however, beyond the scope of this thesis. We refer therefore to [74, 125] for a survey of local optical flow approaches.

2.4 Summary

This chapter provides us with the basis for our approaches we will discuss in the following chapters. We started in this chapter with fundamental properties of the optical flow constraint equation and presented how variational approaches are used for the computation of optical flows. Necessary additional constraints are introduced by appropriate regularisation terms. For example, Horn and Schunck (2.8) regularise the brightness constancy assumption with a spatial smoothness term. The approach of Weickert and Schnörr (2.16) penalised the spatio-temporal variation, leading to a static approach as the entire spatio-temporal image volume is used to determine the optical flow of an image sequence. We consider all the presented optical flow approaches in this chapter as "static": The time-dependent optical flow is computed for a fixed point in time. In contrast to the "dynamic" viewpoint of the motion, which describes the fact that the flow is moving and that the positions of flow particles change with time.

In the following chapters we propose two new approaches which incorporate physically motivated regularisation terms for the optical flow computation. In chapter 3 we introduce a regularisation term that is based on a transport equation (the Burgers equation) which models dynamic image motion. In chapter 4 we follow that idea and obtain an optimal control formulation for motion estimation. Both approaches allow the detection of unexpected movements in image sequences in terms of a corresponding force field. Our main intention is to investigate the properties and the potential of the optical flow approaches that results from the idea to model dynamic image motion by a transport equation rather than to contribute to the accurateness of state-of-the-art optical flow approaches.

3

Dynamic Optical Flow Approach

In this chapter we propose a new variational optical flow approach with a novel PDE-based representation of image motion. The approaches in the previous chapter represent a static viewpoint to image motion processing. In this chapter we extend this viewpoint to a dynamic one, incorporating the Burgers equation as a physical prior to dynamic image motion modelling. Deviations from the expected velocity distribution generate vector fields that may serve as attentional mechanism for a superordinate processing stage. Our approach performs a spatio-temporal regularisation in a recursive online fashion, unlike previous approaches evaluating the entire spatio-temporal image volume in a batch processing mode.

3.1 Introduction and Motivation

Previously, most authors concentrated on spatial smoothness constraints within the optical flow computation (cf. sec. 2.2.2, 2.2.3, 2.3). In real-world applications however, we are often concerned with long image sequences which allows the utilisation of the temporal coherency as well as physical assumptions of the of motion. Some efforts have been made to include temporal smoothness as introduced in section 2.2.4 and in [101]. In these approaches, time is considered as a third dimension analogue to the two spatial dimensions. This lead to a three-dimensional elliptic PDE with a static description of motion, where a flow vector is determined by the solution of a linear system involving the data of the full 3D image volume. Solving this variational problem allows the computation of the optical flow only after the last frame of the image volume has been captured.

Our approach is based on modelling a fictive compressible fluid, using the Burgers equation [23], in combination with a convex regularisation approach to image motion computation [100, 122, 142]. We employed the non-linear hyperbolic Burgers equation, as physical prior knowledge for image motion, which acts as temporal regularisation. The Burgers equation belongs to the governing equations in fluid dynamics and embodies the convection term of the Navier-Stokes equation. This non-linear convective term reads $(u \cdot \nabla)u$ and is also known as inertia term. The physical interpretation of this term is that the velocity u is transported by itself. We emphasise that the inertia behaviour of the Burgers term results in a temporal smooth change of the velocities.

In terms of computer vision, the Burgers equation exploits the most elementary knowledge available in connection with image motion, which can be informally expressed as “structures do not jump”. The physics underlying any motion phenomenon embodies some inertia leading to smooth changes of velocities. Maybe starting with the work of Heel [63, 64], this knowledge has been exploited in countless applications in computer vision during the last two decades, mainly in terms of the Kalman filter, and more recently through its modern extensions [41].

A key property of the Kalman filter is that it applies to systems, where the system dynamics is summarised by a set of variables which are a function of the time alone. Such systems are known as *lumped systems*. When applied to raw pixel intensities for image motion computation [63], for example, or to depth estimates from image sequences [92], this leads to the well-known recursive update equations for computing corresponding estimates at each pixel location, *without* any *spatial* context, however. To achieve the latter, *parametric* models, e.g. models of surface patches in the scene, have to be assumed, which not only requires far more *specific* knowledge about the scene, but also necessitates transferring the Kalman formalism into a corresponding parameter space, typically through non-linear equations relating the directly observable visual measurements to the parameters as new system variables.

Our ansatz, which we explain in detail in the following sections has the following main properties:

(i) We take into account *spatial context* by using the image-driven regularisation term introduced in chapter 2, leading to a *distributed-parameter approach* [1] which states are governed by partial differential equations (PDEs), rather than ordinary differential equations as for lumped systems.

(ii) The approach exploits the aforementioned elementary knowledge in connection with motion computation that moving structures usually exhibit some inertia in its most rudimentary form. That means that velocities are not expected to change. This is incorporated by the Burgers equation.

(iii) The computational structure enables an online motion computation [2].

The approach also admits a *control based interpretation*: if image measurements indicate changes of the current velocity distribution, fictive forces modify the

---

[1] Distributed-parameter system theory is concerned with the dynamic behaviour of processes distributed in space as well as evolving in time [51]

[2] The motion computation can be started as soon as the respective images has been captured.

system state accordingly. The presence of such forces may serve as an attentional mechanism notifying a higher-level processing stage about unexpected motion events.

Our main motivation is to introduce the Burgers equation as a new regularisation term as physically plausible description for motion within a variational framework and to explore the capabilities and properties of the resulting approach. To do so we designed experiments to extract the main properties of our approach. Furthermore, we compare our approach with the well-known Horn and Schunck approach - using standard optical flow image sequences - rather than with sophisticated algorithms that are much more difficult to implement. This allows to detect the fundamental differences of our approach and still allows the reader familiar with optical flow approaches a grading of the accuracy for the resulting flow fields.

3.1.1 Overview

The remaining part of this chapter is organised as follows: First, we present in section 3.1.2 some work that - related to ours - make use of ideas from the field of fluid dynamics for computer vision tasks. Then we present in section 3.2 our variational approach and discuss the impact of the Burgers term which we introduce as regularisation term. Another interpretation of our approach is provided in section 3.2.3 where we shortly speculate about connections to the field of visual perception and attention mechanisms. The most important elements of our implementation are presented in section 3.3. This is followed by experiments in section 3.4, where we investigate the fundamental properties of our approach. Finally, we conclude with section 3.5.

3.1.2 Further Related Work

Concerning property (i) stated above, it is well-known that *non-local* variational approaches, particularly those exploiting *spatio-temporal* context [143], lead to robust estimates of coherent motion patterns. On the other hand, such approaches are *static* in that an entire spatio-temporal image volume has to be processed at once in batch mode. In contrast, our approach is *dynamic* and leads to recursive computations in online mode, while still performing temporal regularisation.

A prominent example for other work exploiting models from fluid mechanics is the work of Christensen on image registration [33] which subsequently instigated a lot of related research (cf. [8] and references therein). The authors also use the momentum equation as prior knowledge. However, while we merely use the material derivative of the velocity (in order to mimic inertia), they consider the opposite special case of very low Reynolds numbers and drop the inertial terms. Furthermore, as we consider relatively small displacements between video frames, we do not distinguish between velocity and displacements.

Other related work concerns the vorticity transport formulation of the *incompressible* Navier-Stokes equation used for image inpainting [9] (as presented in sec. 5.2.1). There are two conceptual differences to our approach: First, their inpainting approach transports a function of the grey value field while we transport the velocity field itself. Second, we utilise a *compressible* model of fluids, because incompressibility almost never holds true for two-dimensional deformations induced by relative motions to a three-dimensional scene.

Further related work is given by Ruhnau, Stahl and Schnörr [118, 117]. They present a variational approach for motion estimation of instationary fluid flows from image sequences. Their regularisation term also incorporate physically plausible prior knowledge: the vorticity transport equation. The optimisation is done in a receding horizon manner contrary to previous approaches to spatio-temporal regularisation.

Finally we point out that while our approach performs spatio-temporal filtering and may be regarded as a rudimentary version of a *distributed-parameter* Kalman filter, it falls short of an exact corresponding mathematical extension (cf., e.g. [17]). The advantage, however, is a deterministic and easy-to-parallelise computational architecture.

3.2 Approach

In the following we explain our variational model where we make use of the notation provided in section 2.1.1. We first introduce the Burgers equation which is used as regularisation term within our variational model.

3.2.1 Regularisation Term: The 2D Burgers Equation

The problems discussed in chapter 2 belong to the class of elliptic problems, which appear through the derivation of the gradient of the flow field (smoothness regularisation term) as second-order derivative terms which express the diffusion phenomenon with a tendency to smooth out gradients (in a homogeneous, isotropic or anisotropic way). In contrast, the inviscid Burgers equation belongs to systems of hyperbolic conservation laws, as convective flux, which appears as first-order derivative term and expresses the transport properties of flow systems. It is represented by the following non-linear partial differential equation

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u = 0, \tag{3.1}$$

with initial data

$$u(x,t)_{t=0} = u_0, \tag{3.2}$$

where $u = (u_1, u_2)^\top$, $u = u(x,t)$, $x = (x_1(t), x_2(t))$ represents a two-dimensional velocity field.

Note, that the Burgers equation appears as convection term not only in the fundamental equations of fluid mechanics, the Navier-Stokes equations, but also in many other time-dependent dynamic equations in physics. The dynamic behaviour of the Burgers equation is described in detail in section 7.5.

The first part of the Burgers equation (3.1), namely the time derivative of u, is the local derivative. It describes the change of the velocity with respect to time at a fixed point in space. In [143], Weickert and Schnörr assume that the velocity at a certain point in space changes only smoothly and therefore penalise deviations from $\partial_t u = 0$. While this assumption is often valid it might fail in two scenarios:

- When fluid flows are analysed, the assumption is admissible as long as the flow stays stationary. However, in scenarios with high Reynold's numbers (for fluid experimentalists, these scenarios are usually the ones of interest), turbulent flow phenomena evolve - and these flows are usually highly non-stationary.
- Also in everyday scenes, penalising deviations from $\partial_t u = 0$ might fail. While in some scenarios (e.g. moving camera in front of static scene), penalising high temporal velocity gradients is admissible, it is a bad idea in other scenarios: If we consider, e.g., moving objects in front of a stationary background, the temporal velocity gradient at the spatial position of the moving objects' boundaries will be extremely large (even-though the objects do not change their speed over time!). Even when using an edge-preserving spatial regulariser, the temporal regularisation will result in blurred object edges.

We want to improve optical flow computation in above scenarios where the simple temporal regularisation of [143] fails. Therefore, we do not merely take into account the local derivative but include the second part of the Burgers equation - the convective derivative of the velocity - for regularisation. Physically, this is the rate of change due to the movement of a fluid element from one location to another in the flow field. It is therefore taken into account, that a fluid element changes its position over time - while maintaining it's velocity, which leads to more accurate motion boundaries.

Here we introduce the Burgers equation (3.1) as a new regularisation term into a variational model. We combine the Burgers term with the approach of Nagel [100] to a dynamical time regularised approach.

3.2.2 Variational Model

The standard variational approaches of Horn and Schunck (2.8) and Nagel (2.15), minimise an objective function over the entire image domain Ω that includes the squared form of the optical flow constraint equation (2.2) along with a regularisation term that enforces smoothness of the image flow in a homogeneous or anisotropic way, respectively.

The idea behind our work presented in this chapter is an attempt to change the static viewpoint of these approaches to a *dynamic* one by exploiting the transport properties of the Burgers equation. Therefore, we solve the Burgers equation (3.1) with initial condition (3.2) for the time interval $[0, T]$ between a subsequent pair of image frames, where u_0 denotes our current motion estimate. We obtain a velocity field $u_T := u(x, T)$ that has been transported by itself. It can be regarded as a *predicted* vector field, based on the constant velocity assumption of the Burgers equation [3]. The matching term that serves as our new regulariser penalises the deviation from the predicted velocity values u_T:

$$J(u) = \frac{1}{2} \int\limits_{\Omega} \left\{ (u \cdot \nabla I + \partial_t I)^2 + \beta \|u - u_T\|^2 \right\} dx. \tag{3.3}$$

While this regularisation term penalises deviations between the current velocity u and the propagated velocity of the preceding frame u_T, it does not enforce spatial smoothness of the velocity, which leads in practise to increasingly noisy velocity estimates. Therefore, we also incorporate the *spatial* regularisation term of Nagel (cf. 2.2.3) and obtain the following objective function:

$$\begin{aligned} J(u) = \frac{1}{2} \int\limits_{\Omega} \Big\{ \Big((u \cdot \nabla) I + \partial_t I \Big)^2 + \alpha \Big(\nabla u_1^\top D(\nabla I) \nabla u_1 + \nabla u_2^\top D(\nabla I) \nabla u_2 \Big) \\ + \beta \|u - u_T\|^2 \Big\} dx. \end{aligned} \tag{3.4}$$

By minimising (3.4), we combine the predicted state in terms of the vector field u_T with current measurements, given by the optical flow constraint equation (2.2), along with a spatial regularisation. The resulting global minimiser $\bar{u}$ defines the initial data u_0 of the Burgers equation (3.1) for the next image pair.

The third term of the energy functional (3.4) incorporates the Burgers equation and yields temporal regularisation. This term relates the velocity information inferred from the corresponding image pair with the assumed temporal prior knowledge induced by the Burgers equation. Our approach adds a temporal correction term to the spatial regularisation approaches described in the previous section. This correction term acts as a fictive force and corrects the optical flow whenever an object in the image sequence changes its moving direction.

The two regularisation parameters $\alpha > 0$ and $\beta > 0$ specify the importance of the second and the third term in (3.4), respectively. Higher values of α lead to

---

[3] In this context, "constant velocity" means that a fictive fluid particle's velocity does not change over time. This, however, does not necessarily imply, that the velocity at a certain location does not change with time. For deeper insights into Eulerian versus Lagrangian motion, we refer to [128, 32].

smoother flow fields if $D(\nabla I) = \mathbf{1}$ or to more direction-dependent smoother flow fields if $D(\nabla I) \neq \mathbf{1}$. Higher values of β result in a stronger inertia behaviour of the flow field as the compliance with the Burgers equation is enforced more strictly - the velocity field therefore adapts much slower to the present motion. In this case the flow field which satisfies the Burgers equation leads to a flow field which adapts slower to the measured motion (cf. sec. 3.4.1).

Following the calculations described in 2.2.1, the minimisation of the convex functional (3.4) leads to the following Euler-Lagrange equations

$$\begin{aligned} 0 &= (I_t + (u \cdot \nabla)I)I_{x_1} - \alpha(\nabla(D(\nabla I)\nabla u_1) + \beta(u_1 - u_{1_T}) \\ 0 &= (I_t + (u \cdot \nabla)I)I_{x_2} - \alpha(\nabla(D(\nabla I)\nabla u_2) + \beta(u_2 - u_{2_T}), \end{aligned}$$

with homogeneous von Neumann boundary conditions ($\partial_n u = 0$). The minimisation of (3.4) is solved in a standard way (cf. sec. 2.2 and 3.3.2 for details).

3.2.3 Further Interpretation

An important property of our approach that incorporates the Burgers equation (3.1) in a variational model (3.4) is that temporal context is taken into account without too much additional computational costs. Rather, the solutions to (3.1) and (3.4) can be recursively computed in an online processing mode. Note that, regarding to the temporal regularisation, we do *not* claim strict mathematical equivalence to the approach [143].

The unique global minimiser $\bar{u}$ of the quadratic functional (2.15) of Nagel is determined by the variational equation

$$a(\bar{u}, v) = b(v) \ , \quad \forall v, \tag{3.5}$$

where the bilinear form $a(\cdot,\cdot)$ and the linear form $b(\cdot)$ comprise all quadratic and linear terms in u of the functional (2.15), respectively. A physical interpretation regards (3.5) as stationary (deformation) state of the “elastic” image domain Ω, defined as equilibrium between the internal deformation energy $a(u,u)$ and external forces $b(u)$, according to the principal of virtual work. Adopting this viewpoint, the additional penalty term in (3.4) adds an additional internal energy term $\|u\|_{L^2(\Omega)}$ regularising the degenerate data term, and an additional external force term $\langle u_T, u\rangle_{L^2(\Omega)}$ related to the temporal prediction.

Another interesting property arises if we interpret our approach from the viewpoint of visual perception. Our approach (3.1) and (3.4) allows to match computed motions against expectations without making *specific* assumptions about the underlying scene (geometry, rigid/non-rigid, etc.). As a consequence, significant deviations from expected motions may be used to focus the attention of a superordinate processing stage. In section 3.4, we will inspect this aspect in terms of the vector fields

$$f(x) := \big(u - u_T\big)(x), \tag{3.6}$$

which explicitly expresses the deviation from the expected motion field.

3.3 Implementation Overview

In this section we summarise how we implemented our approach described above. Here we mainly provide an overview of our implementation and refer for details to the appropriate sections in the numeric chapter.

The implementation of our approach involves the solution of the two-dimensional Burgers equation (3.1) and the minimisation of (3.4). Both steps are decoupled and can be computed separately leading to an efficient computation. From these two computations, the numerical implementation of the non-linear Burgers equation is the more intricate task. In the following we explain the solution of our variational approach (where we refer to particular numerical sections).

3.3.1 Discretisation of the Burgers Equation

Basically, to solve the time-dependent Burgers equation (3.1), where we consider von Neumann boundary conditions, we use a second-order conservative finite difference method. Due to the discretisation, fluxes are numerically computed by solving the appropriate equations at pixel edges where all fluxes are evaluated and differenced at the same time. The physical reasonable behaviour at discontinuities (like shocks) is obtained by using solutions of the appropriate Riemann problem and by applying Van Leer limiters to second-order terms. For details, we refer to section 8.3 as we have to cope with over- and undershoots, with different types of shock formations, with the compliance of conditions (entropy-, monotony-, CFL-condition, etc.) and different discretisation schemes.

3.3.2 Variational Approach

We use piecewise linear finite elements to discretise the variational equation (3.5), and compute the global minimiser $\bar{u}$ by solving the resulting linear system of equations. We can specify the variational equation explicitly as:

$$a\big((\bar{u}_1, \bar{u}_2)^\top, (v_1, v_2)^\top\big) = b\big((v_1, v_2)^\top\big) , \quad \forall v_1, v_2 , \tag{3.7}$$

where

$$
\begin{aligned}
a(\bar{u}, v) = \int_\Omega \Big\{ & \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}^\top \nabla I \nabla I^\top \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \\
& + \alpha \Big(\nabla u_1^\top D(\nabla I) \nabla v_1 + \nabla u_2^\top D(\nabla I) \nabla v_2 \Big) + \beta u^\top v \Big\} dx \qquad (3.8) \\
b(v) = \int_\Omega \Big\{ & - \partial_t I \nabla I^\top v + \beta \, u_T^\top v \Big\} dx. \qquad (3.9)
\end{aligned}
$$

The simplest discretisation is obtained by choosing a regular triangulation of the image domain Ω and attaching to each pixel position a piecewise linear basis function $\phi(x_1, x_2)$ for each function u_1, u_2, v_1, v_2, as we already illustrated in section 2.2.

The resulting linear system is sparse and positive definite. Thus the vector components u_1, u_2 can be conveniently computed by some corresponding iterative solver as described in section 9.1. Note, that the natural occurring von Neumann boundary conditions are automatically encoded by the finite element discretisation.

3.3.3 Coarse-to-Fine Approach and Iterative Registration

In order to handle large motions in image sequences we incorporate a coarse-to-fine technique in combination with an iterative registration approach as described in section 9.2 within the numerical part of this thesis.

3.4 Experiments

With the experiments of this section, we intend to examine and illustrate the key properties of our approach and explore the potential benefit of it. We evaluate sequences that contain uniform and non-uniform motion patterns and compare our results obtained for ground-truth image sequences with homogeneous (Horn and Schunck (2.7)) and image-driven (Nagel and Enkelmann (2.14)) regularisation techniques. Furthermore, we present some results for a real-world image sequence.

Before we actual show and discuss our experiments we first list where the focus of our approach lies. Then we define which error measure we use for comparison and provide the value ranges for a reasonable parameter selection.

Properties and Aspects

Within our investigation we focus on experiments that are designed to show certain key properties and aspects of our approach. In particular we are interested in

- the inertia behaviour and
- the temporal regularisation

of our approach. Therefore, we investigate the behaviour for image sequences that show the following motion patterns:

- Expected motion, like linear continuous motion
- Unexpected motion events, like abrupt starting or stopping movements, or a sudden change of the motion direction
- Corrupted motion, i.e. image sequences with temporarily missing image data in connection with expected and unexpected motion events

Furthermore, we present how the approach copes with the following situations:

- Uniform and non-uniform motions involving translation, rotation and scaling (zooming)
- Occlusion
- Noisy image data

We discuss these aspects in the following sections.

Error Measures

For a quantitative comparison of our computational results we use the *average angular error (AAE)* [7]

$$AAE(u_o, u_e) = \frac{1}{|\Omega|} \int\limits_{\Omega} \arccos\left(\frac{u_o \cdot u_e}{|u_o||u_e|}\right) dx, \tag{3.10}$$

where $|\cdot|$ denotes the Euclidean norm, $u_o = (u_{o_1}, u_{o_2}, 1)^\top$ the original optical flow vectors, and $u_e = (u_{e_1}, u_{e_2}, 1)^\top$ the estimated optic flow vectors. Note that the time dimension is set to 1 corresponding to the distance of one frame. This measure is currently used as a kind of standard to provide accuracy measures for optical flow results. A comparison with the *root mean square error (RMSE)*

$$RMSE(u_o, u_e) = \frac{1}{|\Omega|} \int\limits_{\Omega} \sqrt{(u_o - u_e)^2} dx \tag{3.11}$$

shows (cf. fig. 3.16 and fig. 3.17), that essentially the same problematic areas are revealed when both error images are plotted. Note that we employ both measures.

For a qualitative evaluation of the obtained flow fields we use the intensity and colour code depicted in figure 3.1. The colour of a pixel indicates the direction of the appropriate vector while the brightness represents its magnitude, as shown for example in figure 3.15 later in this section. The last image in the top row depicts the ground truth velocity field and the first image in the bottom row shows the corresponding colour representation. The colour code reveals much more motion details compared to the raw vector field. The colour code images are scaled with respect to the largest vector that occurs in the vector fields we intend to compare. The coloured border around these images just indicates the colour code for the motion direction.

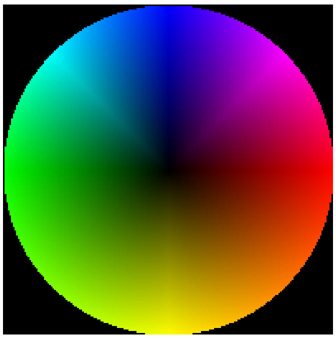


Figure 3.1: Colour code for the vector field representation. The direction is colour coded and the magnitude is represented by the brightness.

Even if our main aim in this thesis is to provide insight into the characteristic behaviour of our approach we compare our results using standard optical flow sequences by the means of the AAE and the RMSE with the results obtained by the Horn and Schunck approach (2.8).

Parameter selection

Our approach comprises the two parameters α and β that have to be determined. The parameter α is a regularisation parameter ($0 < \alpha \in \mathbb{R}$) that controls the degree of spatial smoothness. A larger value leads to a smoother flow field, whereas the smoothness decreases with decreasing α. The parameter β weights the influence of the Burgers equation. Therefore, higher β values results in an optical flow field which amplifies properties comprised in the Burgers term. For example, velocity fields are forced to follow previous motion directions.

Our experiments show that the best choice for these parameters depend on the particular problem (e.g. noise), but are constraint to certain ranges. In our experiments the range of the α-values is between 0.05 and 0.005 (higher values for noisy sequences), while the β-values lie between 0.01 and 0.00002 (higher values for larger inertia effect). For an unknown image sequence a reasonable start setting would be $\alpha = 0.01$ and $\beta = 0.001$. For an illustration we refer to section 3.10 and table 3.1.

3.4.1 Inertia and Motion

In this section we concentrate on the inertia behaviour of our approach by investigating how the approach deals with a sudden change of the motion direction. We show and interpret the results of two experiments. In both the main linear motion changes its direction abruptly from a horizontal to a vertical motion. The difference is that in the first experiment the moving object has a fixed orientation (uniform motion) while in the second experiment the object itself rotates in addition to the main motion (non-uniform motion). We have chosen the Burgers regularisation parameter to be $\beta = 0.005$ which clearly shows the inertia effect on the computed motion field.

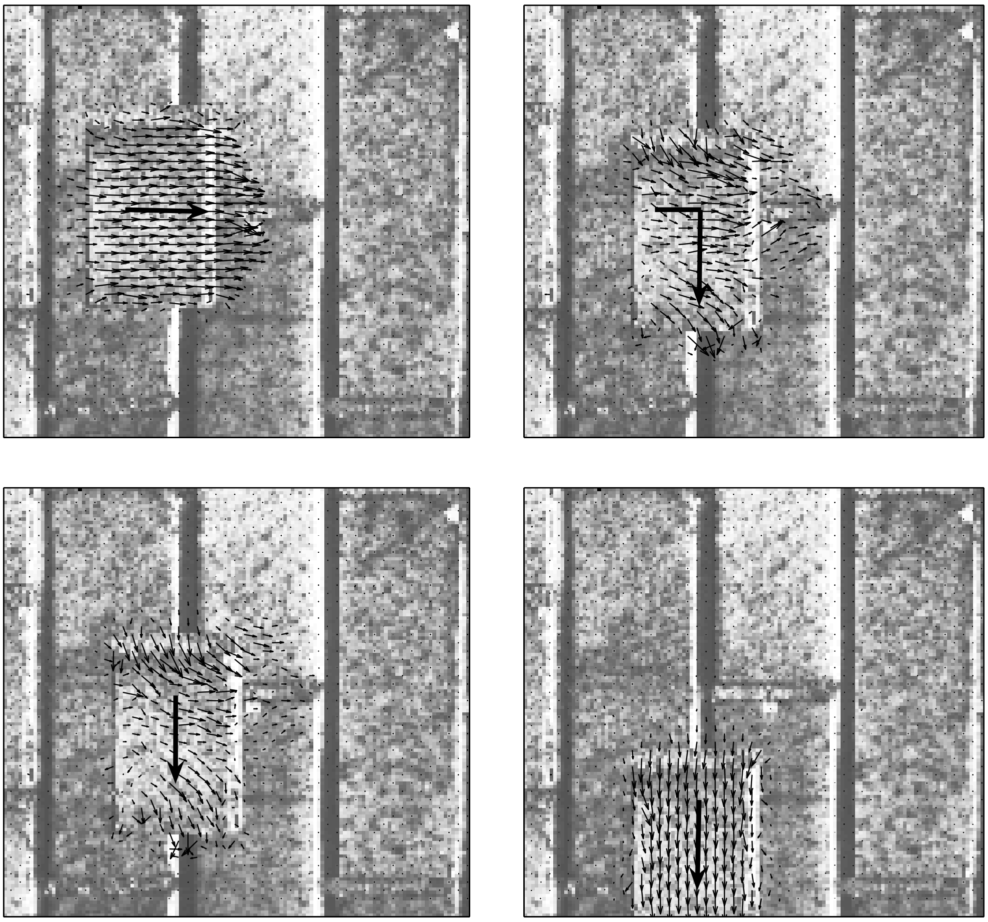


Figure 3.2: Uniform motion and inertia effect: A moving object which abruptly changes the motion direction, as indicated by the large arrows. The figure illustrates the influence of the inertia term of a fictive fluid. Shortly, after the abrupt motion change (top right) the flow tends to keeps its motion direction before adapting to the new direction. See text for further discussion (parameters: $\alpha = 0.01$, $\beta = 0.005$).

The first experiment we conduct is a simple image sequence, where an object (stone) is moving with 7 pixels/frame in front of a structured background and then changes abruptly its motion direction. Figure 3.2 illustrates this sequence along with the effect of the Burgers equation (3.1) on motion computation. The obtained optical flow fields are shown at different times along with the appropriate image frames. The motion of the object is indicated by a large arrow. The upper-right panel depicts the frame directly after the abrupt change of the object's moving direction. Due to the inertia effect, the moving "vector particles" tend to keep their current motion. After a couple of frames (lower-left panel), a new moving front has established which drags the remaining field towards the novel direction. We would like to point out that this effect may be conceived as that of a fictive material, provided with some inertia, which is trying to follow the apparent motion in an image sequence.

This inertia behaviour is not constraint purely to uniform motion. To show that, we perform an analogous experiment for a non-uniform motion pattern. The appropriate "rolling disc" image sequence is depicted in figure 3.3. The motion in this sequence consists of two superimposed motion patterns, a clockwise rotation (5° per frame) and a translation of 3 pixel per frame. The horizontal motion chances abruptly to a vertical one.

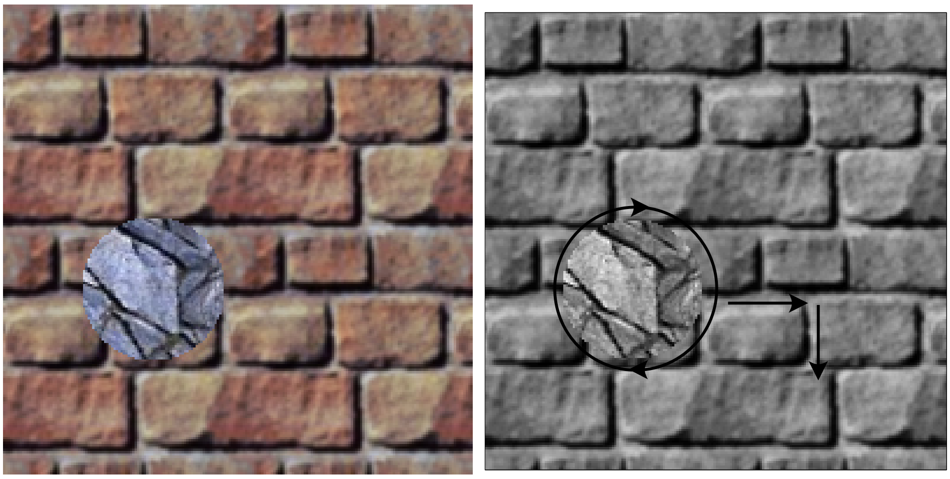


Figure 3.3: "Rolling disc" image sequence. The sequence consists of two superimposed motion patterns, a clockwise rotation (5° per frame) and a translation of 3 pixel per frame. The horizontal motion changes abruptly to a vertical motion. For a better discrimination of the moving object and the background we depicted (**left**) a colour version of the first frame of the sequence. **Right:** First frame of the grey value image sequence, which was used in our computations. The arrows indicate the movement of the disc.

Again, we observe (now for a non-uniform motion pattern) a "temporal filtering effect" due to the inertia term of the Burgers equation (3.1) as depicted in the upper-right panel of figure 3.4. As in the previous experiment directly after an abrupt change of the object's moving direction the moving particles tend to keep their current motion as a consequence of inertia. During the next frames as shown in the lower-left panel, a new moving front has established and the vector field adapts to the novel direction.

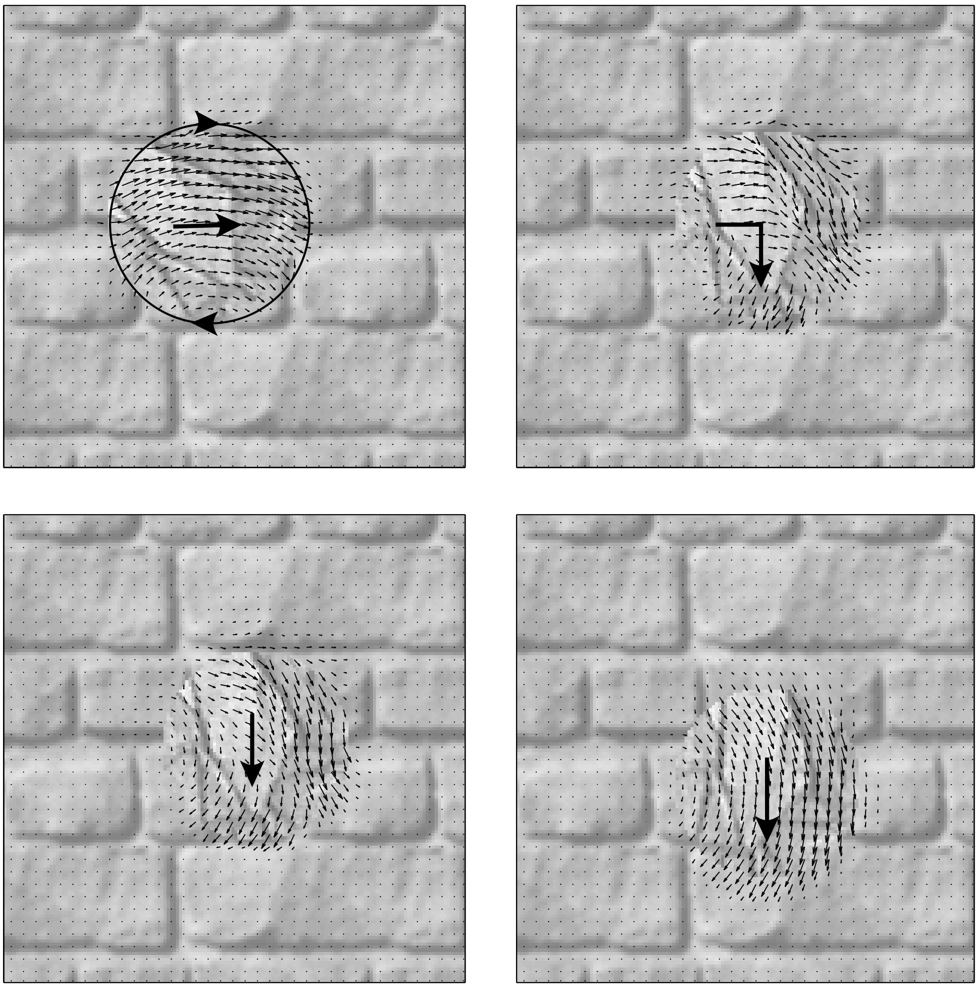


Figure 3.4: Inertia effect for a non-uniform motion: An abrupt change in the main motion direction of a rotating disc, as indicated by the large arrows illustrates the influence of the inertia of a fictive fluid which is trying to adapt to the apparent image motion. The vector field tends to keep its direction before adapting to the new one. See text for further discussion (parameters: $\alpha = 0.01$, $\beta = 0.005$).

Note, that we do not regard this temporal period of “confusion” in these experiments (uniform and non-uniform motion patterns) as a drawback. Rather, in cases where the observed motion is conform with the expectation, the inertia enables the fictive fluid to exert a temporal filtering effect, as experimentally demonstrated above and the message “nothing interesting happens” can be signalled to a superordinate processing stage. On the other hand, in situations with a sudden change of motion the discrepancy between observed and expected motion may be used to generate a message so as to focus the attention of superordinate visual processing stage. This aspect is further investigated in the next section.

3.4.2 Focus of Attention and Motion Compression

In this section we show that our optical flow approach can be used as an attention mechanism to detect unexpected motions. In order to do so we designed a simple image sequence that contains a stopping and starting event of an otherwise constant moving object. The results we obtain for these two events are shown in figure 3.5. But note, in this case the depicted vector fields are *not* the optical flows but the vector fields f defined in (3.6) which quantify deviations from the system's expectation.

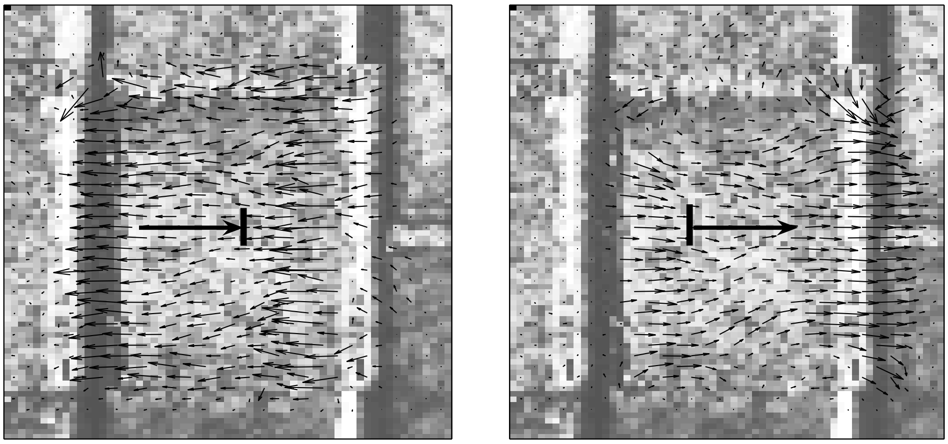


Figure 3.5: **Left:** Vector field f when the object suddenly stops. **Right:** Vector field f (3.6) when the object starts moving again. See text for further discussion (parameters: $\alpha = 0.01$, $\beta = 0.001$).

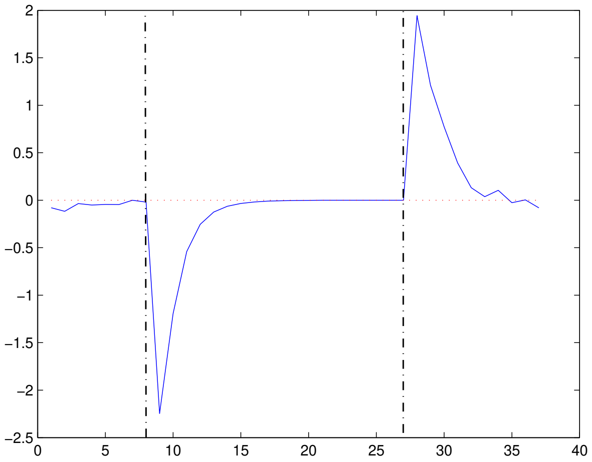


Figure 3.6: Mean of the vector field f (y-axis) over 37 frames (x-axis) corresponding to the image sequence shown in figure 3.5. The dashed lines mark when the object stopped and started, respectively. Signal peaks could be used to focus attention of a superordinate processing level to these events (parameters: $\beta = 0.001$, $\alpha = 0.01$).

For the stopping event (left panel) the vector field points into the opposite direction of the object movement. This is due to the “negative acceleration” of the object. For the starting event (right panel) the acceleration and the object movement point into the same direction.

In order to illustrate how f enables us to detect ”interesting” events we plot in figure 3.6 the mean value of f (computed for the horizontal direction over the object region) as a function of time (frame number). The two vertical dashed lines indicate when the object stopped and started, respectively. For both events, the curve shows a strong signal peak with a delay of one frame. In principle, using for example a simple threshold, this peaks could be used to focus the attention of a superordinate processing stage to these events.

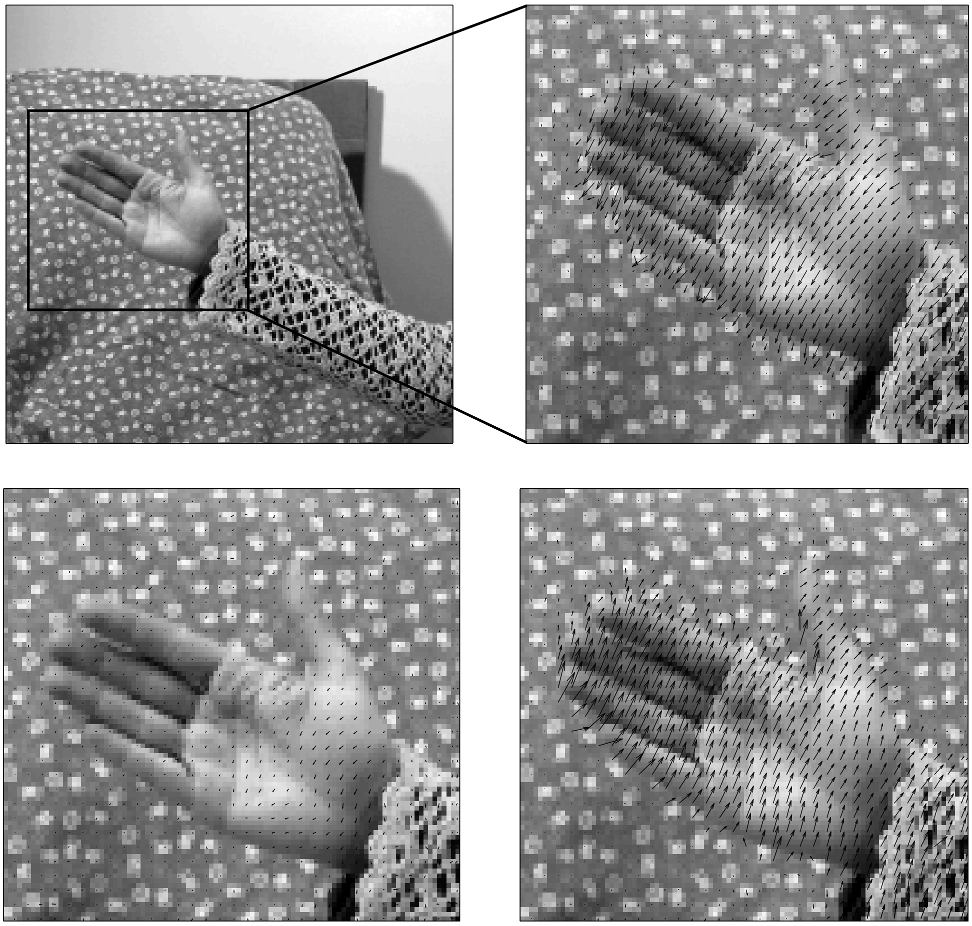


Figure 3.7: Top: A waving hand and its estimated optical flow (right) computed by (3.4). **Bottom:** At a turning point, the estimated velocity field is nearly zero (left) while the deviation from the expected motion is maximal, causing a “negative” vector field f (right) (parameters: $\alpha = 0.005$, $\beta = 0.001$).

We illustrate this aspect in figure 3.7 for a real image sequence showing a waving hand. The turning points of the hand lead to periodic peaks of the vector field f. In between, when the hand moves continuously the motion can be predicted well (f is nearly zero), rendering any communication to an external module unnecessary.

The shown experiments indicate that our optical flow approach naturally includes the easily extractable knowledge of interesting motion events.

3.4.3 Missing Image Measurements and Occlusion

Missing Image Measurements

Let us consider the behaviour of our approach when entire subsequent image frames are missing within an image sequence. These experiments are designed in order to explicitly demonstrate the transport behaviour to the Burgers term. As soon as no further image information is available the vector field is mainly transported by itself. These behaviour can also be exploited to follow moving objects that disappear behind other objects. We investigate this again for uniform and non-uniform motion patterns.

Figure 3.8 shows a sequence of an uniformly moving object where after some time three frames are missing. As the motion coincides nearly with the expectation implicitly made by the Burgers term the motion prediction is exact as well, and the losses of image data is immaterial and can be compensated. We obtain reasonable flow fields even for the missing frames. Note, that purely spatial approaches considering only two frames - as described in chapter 2 - can not deal with such a situation. However, the spatio-temporal approach of Weickert and Schnörr (2.16) provides - due to the three-dimensional smoothness regularisation term ∇_θ - an average flow field for the missing image. We compare the results obtained for both approaches below and explore the features of our new regularisation term along with a qualitative evaluation.

Therefore, we will describe the appropriate experiment in more detail: The top row of figure 3.9 depicts 5 consecutive image frames of the image sequence that is indicated in figure 3.8. The first frame is the last image where a measurement is still available. The following three frames contain no image information, while the last frame provides image data again. The qualitative evaluation of the flow results for both approaches are shown in the rows below. The second row in 3.9 shows the colour code representation for the best optical flow fields for the five image frames obtained by the approach of Weickert and Schnörr (2.16). The remnant flow field are clearly the result of the temporal smoothing leading to the fading flow in the middle frame. The parameter α is set to 0.01.

The third row contains the velocity estimates computed by our approach (3.4) with $\alpha = 0.01$ and $\beta = 0.01$. We observe clearly that without image information the velocity field is still transported by itself mimicking the motion of the moving object.

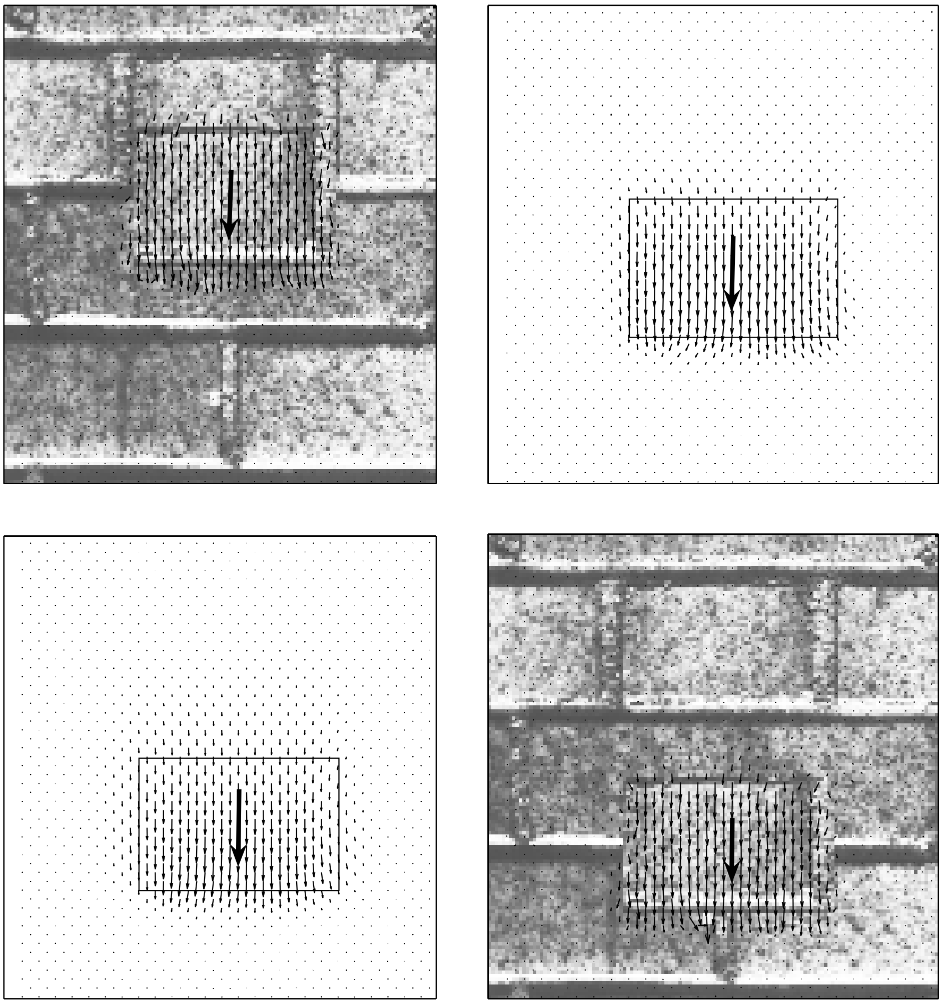


Figure 3.8: Estimated motion of an uniformly moving object while no image measurements are available. The second and third image show the flow field estimates for missing image frames. The inertia of the fictive material allows a reasonable motion prediction, loss of data is compensated (parameters: $\alpha = 0.01$, $\beta = 0.001$).

The motion boundaries are much better preserved than by the spatio-temporal approach (2.16). The fourth row is essentially the same experiment but effectively switching off the spatial regularisation during the period of data loss. The shock front in the moving direction is sharper but more-vulnerable against perturbations of the velocity estimates leading to a non-smooth motion boundary.

However, this experiment shades light how the Burgers term predicts uniform motion by the transport process leading to a temporal filtering effect that is superior to the spatio-temporal modelling of (2.16).

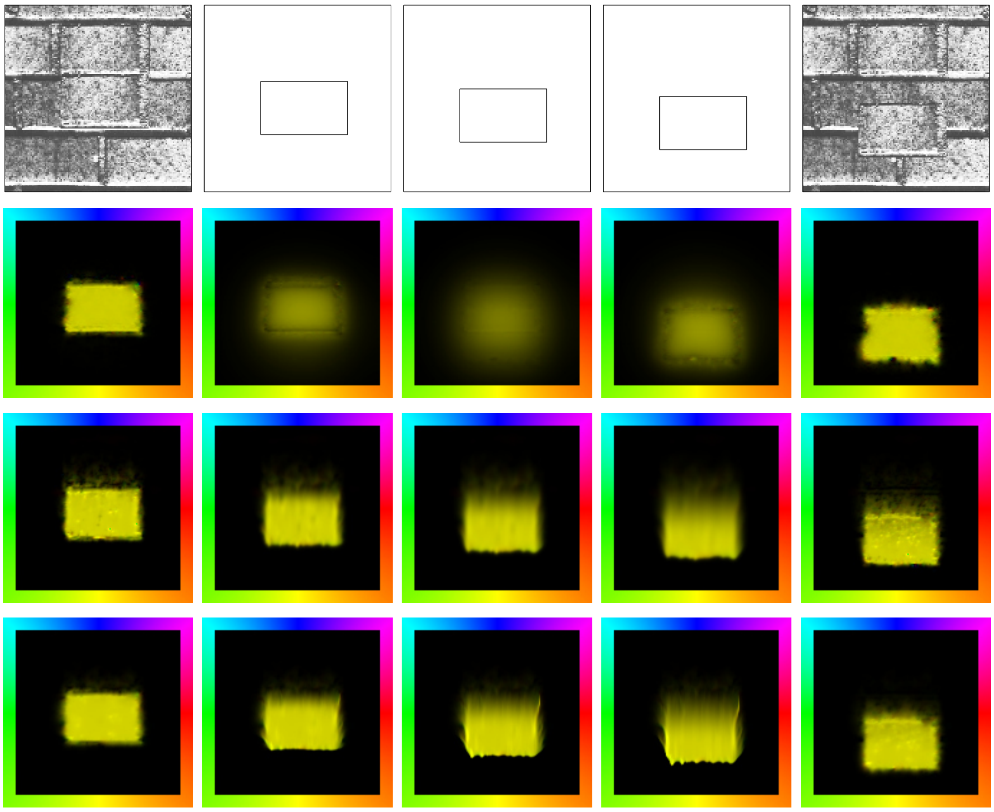


Figure 3.9: Comparison of the temporal regularisation between our approach (3.4) and the spatio-temporal approach (2.16) of Weickert and Schnörr during loss of image measurements. **First Row:** Part of an image sequence with an uniformly moving object where the image information for three frames is completely missing. **Second Row:** Colour code representation for the optical flow estimates of the Weickert and Schnörr approach (2.16), which results in averaged flow fields for the missing image frames (RMSE = 0.0902). **Third Row:** Flow field results for our approach (RMSE = 0.0521), (parameters: $\alpha = 0.01$, $\beta = 0.01$). **Last Row:** The spatial regularisation parameter α is switched off. These experiments demonstrate the capability of the transport process to mimic the object motion during the time of missing data (RMSE = 0.0425).

In our next experiments we explore the explicit influence of the β parameter for the same experimental setup as in figure 3.9 (top row). In figure 3.10 the spatial regularisation parameter is fixed to $\alpha = 0.03$. Each row represents the results we obtained for a different β parameter that deceases from 0.01 to 0.0001. One can observe that with decreasing β parameter the transport capability is reduced until nearly no flow field is obtained for frames with missing image data. This is clear as $\beta = 0$ in our approach means that it is reduced to the approach of Nagel (2.15).

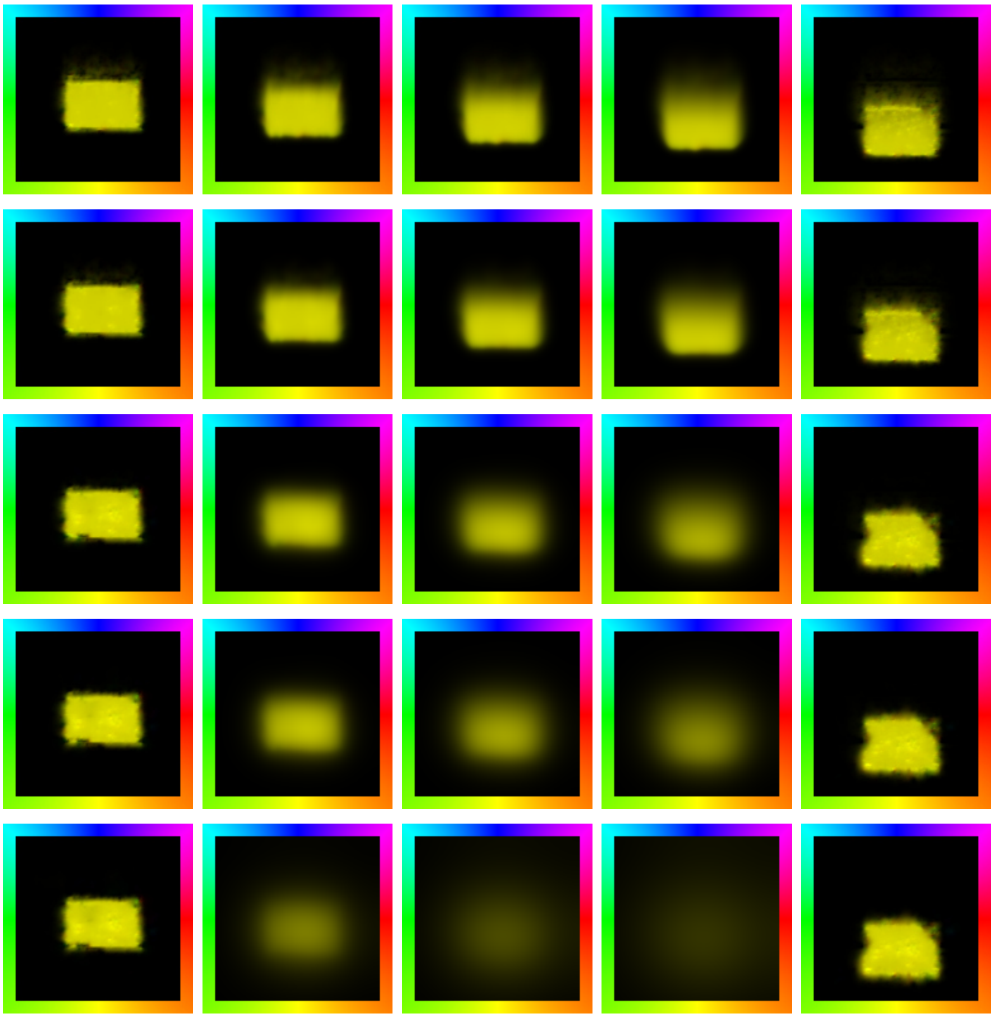


Figure 3.10: Influence of the Burgers equation as physical prior. Each row represents the results for a different β parameter. The spatial regularisation parameter α is fixed to 0.03. The results are shown for the same experimental setup as in figure 3.9 (first row). The parameter is set to $\beta = 0.01$, 0.005, 0.001, 0.0005, and 0.0001, respectively. During the period of data loss we observe that with the decreasing influence of the Burgers term the velocity field gets smoother and loose its silhouette until it nearly fades out completely (last row of fig. 3.10).

The Burgers equation is known to represent a model for shock formations in physics (cf. sec. 7.5). Exploiting this property for computer vision tasks allows to model sharper motion boundaries. The shock front plays the role of the motion boundary (in flow direction) of the moving object. Therefore, we obtain at the motion boundary (fig 3.9) a more accurate flow field as at the motion boundary against the flow direction. Such a scenario is realistic in connection with occlusions or technical limitations in the recording process that lead to data loss.

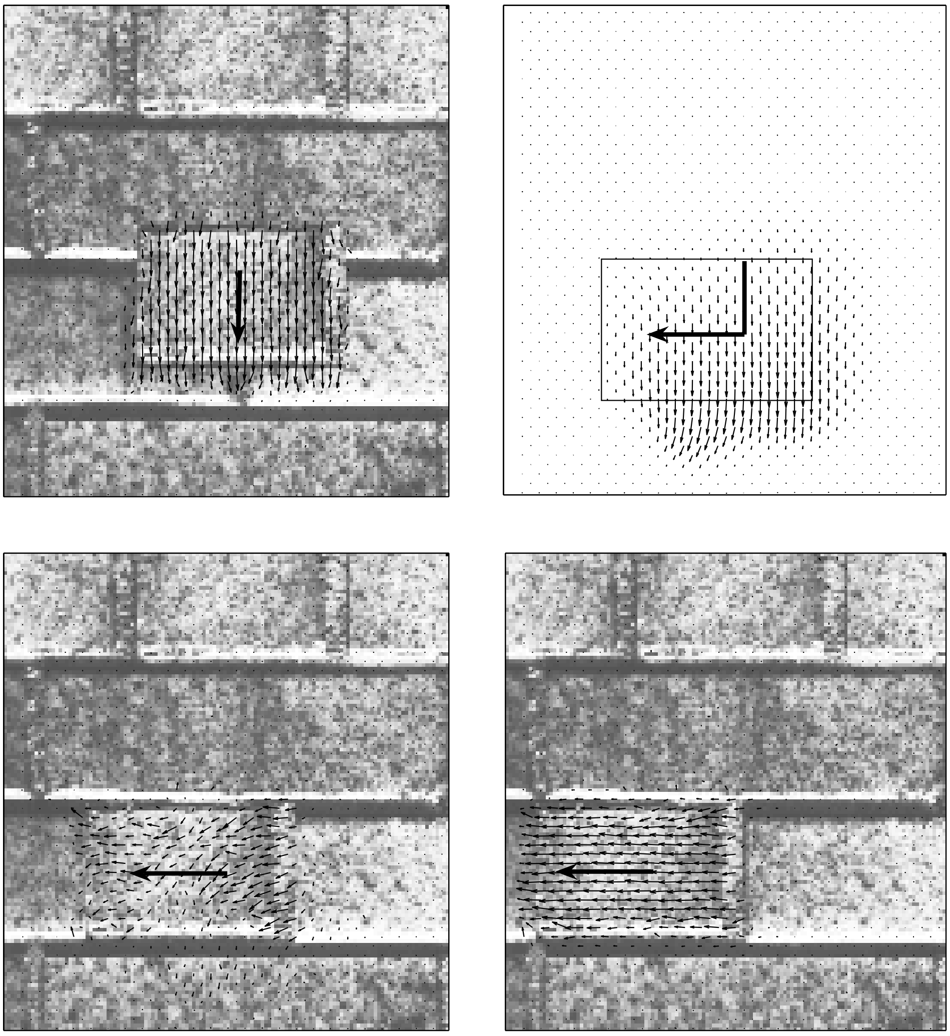


Figure 3.11: The rigid object changes its motion direction during a time of missing image frame data. The flow vectors below the actual object position in the lower-left panel illustrate how the process persists in looking for the moving object in the originally expected direction (parameters: $\alpha = 0.01$, $\beta = 0.001$).

We changed the experimental setup in order to follow the question how our approach can cope with the situation when the object changes its motion direction during the absence of image data. The upper-right panel of figure 3.11 shows that - like in the previous experiment - the algorithm assumes a continuation of the uniform motion of the object. But when the image data is available again (lower-left panel), the velocity direction is updated accordingly. Notice, however, that there are erroneous velocity estimates (below the object) showing that the algorithm is still "searching" the moving object in the originally expected direction.

In the following we perform a similar experiment but now involving a non-uniform motion pattern. The "rolling disc" image sequence we use for this experiment is indicated in figure 3.12 with three missing image frames. Again, our approach has to deal with missing image frame data during which the rotating object changes its direction. Due to the previously observed temporal filtering effect the velocity estimates still mimics the original motion to some degree for the period of missing information. And again when the image measurements of the object are available again the velocity estimates adapt under the influence of inertia to the new motion direction. Nevertheless, we observe that the inertia behaviour of non-uniform vector fields allows only - compared with an uniform vector field - a less accurate interpolation of the real motion.

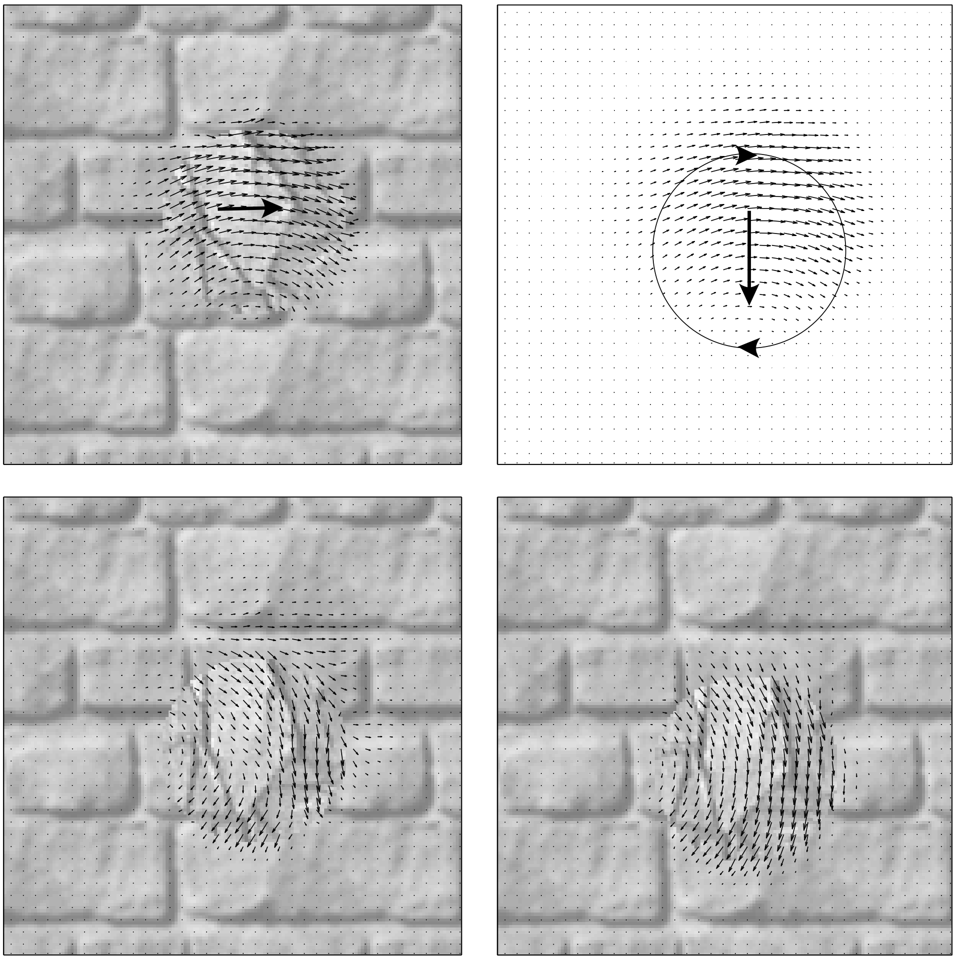


Figure 3.12: Missing image measurements and non-uniform motion: The "rolling disc" changes its motion direction during the time when image data is unavailable. (parameters: $\alpha = 0.01$, $\beta = 0.001$).

From the experiments above we conclude that the transport property of the Burgers equation is in particular suitable for uniform motion patterns to made reasonable predictions of the movement. We exploit this behaviour within our next experiment where we try to predict the motion of an object that vanishes behind a foreground object.

Occlusion

In order to investigate how our approach can cope with occlusion we generated a simple image sequence where a rigid disc (a coin) is moving in vertical direction (uniform motion) and is then occluded by a bar. Note that the coin disappears completely behind this bar and appears again.

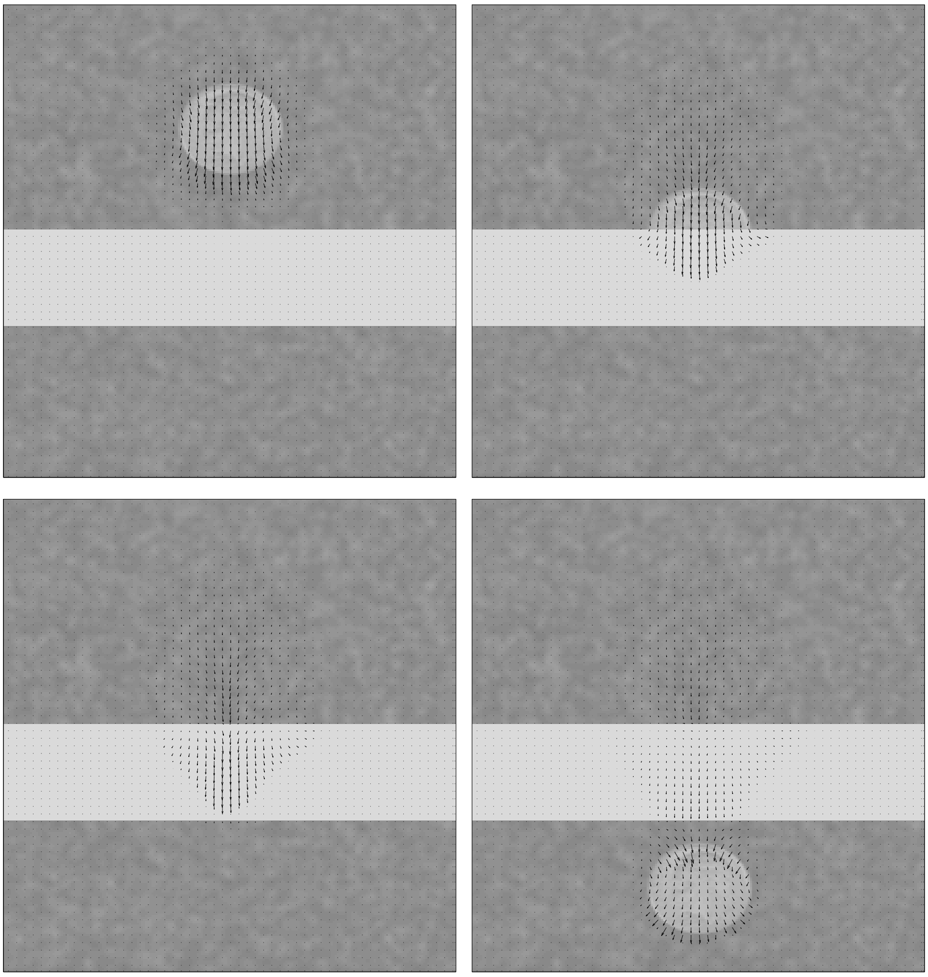


Figure 3.13: Occlusion and the inertia behaviour of the Burgers equation. A coin moves uniformly vertical and disappears behind a bar and appears again. **Top left:** Computed flow field for the second image frame. **Top right:** The coin is half occluded and the shape is still recognisable. **Bottom left:** The coin is entirely occluded by the bar and the velocities are transported by itself without any correcting image measurements. **Bottom right:** When the coin reappears the measurements drag the flow field in the expected direction (parameters: $\alpha = 0.008$, $\beta = 0.01$).

Our result for this sequence is shown in figure 3.13. We observe that for several frames after the coin starts to disappear the shape of the coin is well preserved. But with progressing time the round shape of the vector field dissolves (bottom

left). This is a consequence of the inertia behaviour of the Burgers equation (which is also known as the convection term) which transports velocities with itself. This shows that our approach is capable - up to a certain degree - to deal with occlusions. It is limited in the way, that the velocity field - in absence of the moving object (coin) - is not shape preserving. The transportation process of the Burgers equation leads to the effect that the motion field of an object is smoothed out in the rarefaction area of the flow. Without image information of the moving object the motion boundaries are no longer corrected by the data term of the variational approach (3.4) and are purely determined by the transportation of the velocities.
This experiment illustrates to which degree our approach is capable to cope with occlusions.

3.4.4 Translation, Rotation and Zooming

In this section we provide an evaluation of our approach on the basis of some well known synthetic image sequences for which the ground truth motion data is available. To allow for a quantitative comparison we provide the results we obtain for the Horn and Schunck as well. The image sequences we use show global motion patterns such as rotation, translation and divergence. Note, that we selected for this experiments a homogeneous spatial regularisation within our approach in order to point out properties of the Burgers regularisation at motion boundaries.

In particular we evaluate our approach on the grey value versions of the following three image sequences: the "office", the "sphere", and the "street" sequence [93]. Usually we present our results in the following way: After an indication of the concerned image sequence the true motion field is shown as vector field and in colour code representation. For a qualitative comparison the colour code representation of our dynamic optical flow approach and of the Horn and Schunck approach are presented. After that we provide quantitative error results and finally, we present a graph that shows how the error develops over the entire image sequence.

Zooming - Divergent Motion: "Office" Sequence

The input data for our first experiment is the "office" sequence. This sequence depicts an office scene and consists of 60 frames of size 200x200 pixels. The global divergent motion pattern is induced by a camera zooming into the scene. The optical flow field that we obtain with our approach is shown in figure 3.14.

The results we obtain for our dynamical optical flow approach and the Horn and Schunck approach are presented in figure 3.15. The middle image of the bottom row shows the colour representation of the vector field obtained by our approach and right from it the Horn and Schunck result. Note that the parameters for both approaches have been optimised. Our main observation is that both approaches result in comparable vector fields.

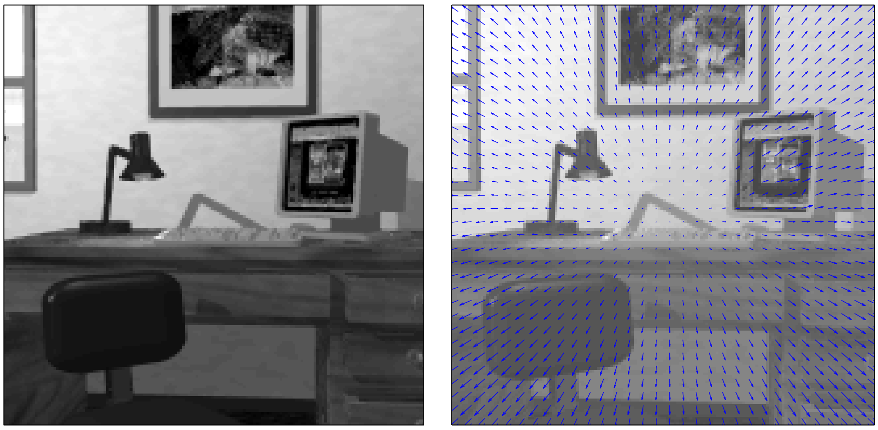


Figure 3.14: Synthetic "office" sequence. The motion is induced by a camera zooming into the scene. Computational results of the dynamic optical flow approach. **Left:** Single frame of the office sequence. **Right:** Subsequent frame of the sequence along with the estimated optical flow field between the shown frames (parameters: $\alpha = 0.01, \beta = 0.001$).

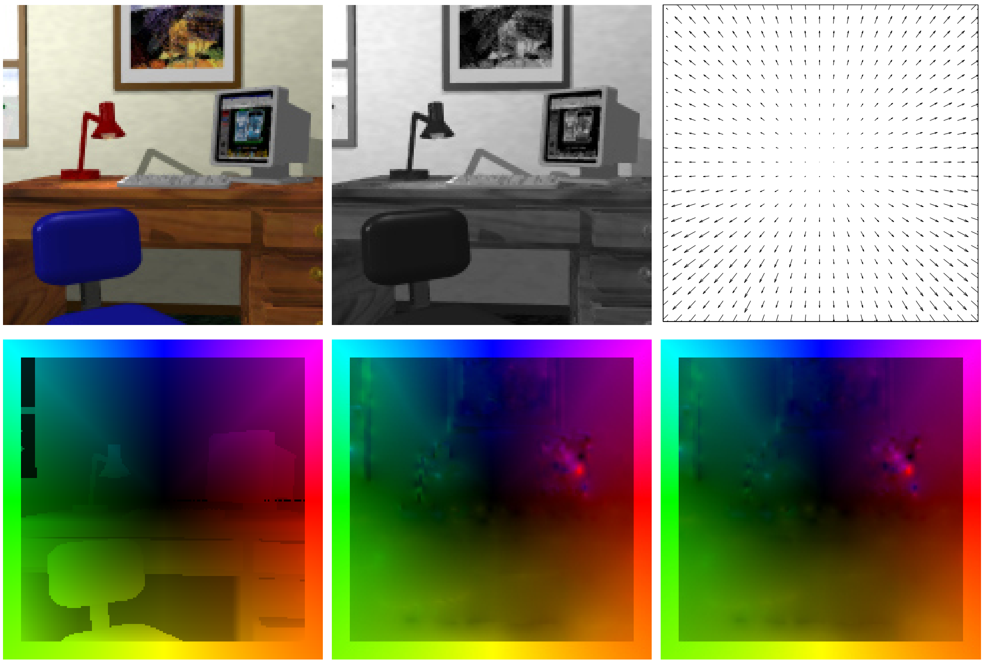


Figure 3.15: Evaluation of the computed flow fields for a divergent motion pattern. **Top left:** Frame 5 of the "office" sequence. **Top centre:** Gray value version of the same frame. **Top right:** Ground truth vector field. **Bottom left:** Corresponding colour code representation of the ground truth vector field. **Bottom centre:** Optical flow field obtained by the optical flow approach (3.4) for dynamic image motion (parameters: $\alpha = 0.05$, $\beta = 0.0006$). **Bottom right:** Optical flow field computed with the Horn and Schunck approach (2.8) with homogeneous regularisation (parameter: $\alpha = 0.04$).

However, the homogeneous regularisation term of the Horn and Schunck approach tends to smooth out motion boundaries more than our approach. This effect is visual near the image boundary where the window can be seen and at the contour of the lamp.

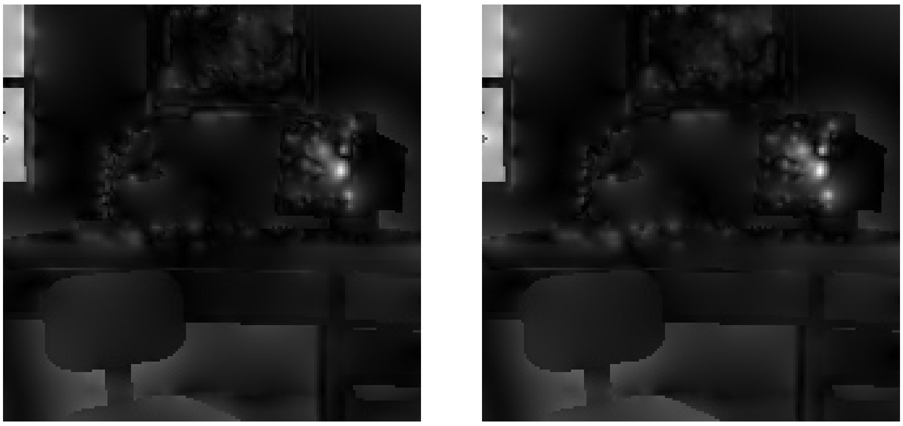


Figure 3.16: Average angular error plots for the ”office” sequence. **Left:** AAE-plot for our result for frame 5 (AAE = 3.180°). **Right:** AAE-plot for the Horn and Schunck result (AAE = 3.323°).

In order to detect which regions turn out to be difficult for the optical flow approaches we provide error plots that show the difference between the estimated and the correct vector field. In figure 3.16 and 3.17 the average angular error and the root mean square error for the velocity estimate between frame 5 and 6 of the ”office” sequence are shown. Both error measurements show essential the same and we restrict our self to present only the AAE plot in similar plots but we still use the RMS error for other comparisons.

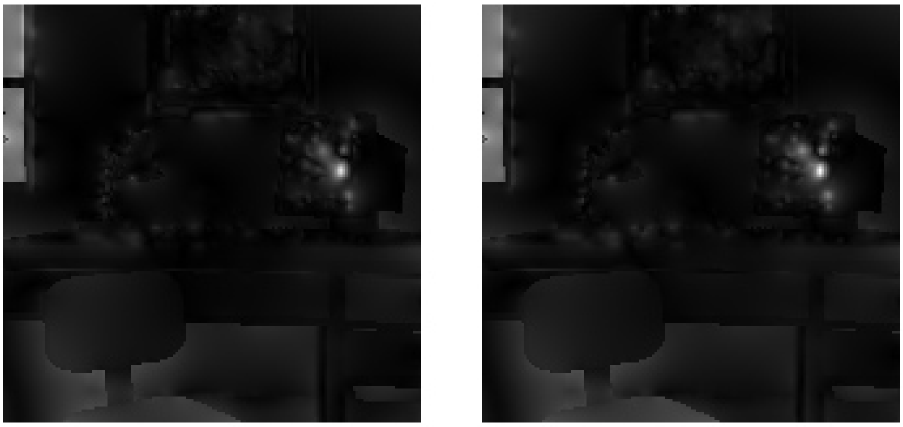


Figure 3.17: Root mean square error for the ”office” sequence. **Left:** RMSE-plot for our result for frame 5 (RMSE = 0.068). **Right:** RMSE-plot for the Horn and Schunck result for the same frame (RMSE = 0.071).

In these plots we observe that motion boundaries are more difficult to extract compared to (slightly textured) homogeneous regions of the objects. However, although we showed that the small influence of the Burgers term for the best parameters results in no significant difference for this sequence (HS-AAE =3.32°, Dyn-AAE = 3.18°). In untextured regions - like the sky within the window - the smoothness term in both approaches lead to dense outward directed flow fields.

The region with the highest error in both error plots is the PC screen. This region is difficult as it consists of fine structured image patterns which appear like noise within the image sequence.

The development of the RMS error over the entire sequence is shown in figure 3.18. With increasing zoom factor and along with an increasing apparent motion the absolute errors increase for both approaches. Furthermore, we note that the divergent motion violates the OFC equation (2.2) (this can be seen at boundary pixels which no longer have corresponding pixel in the following frame). Allover we observe a comparable performance of both approaches. The divergent motion field shows an acceleration from the centre to the boundary while the Burgers equation models a transport process without acceleration. This experiment shows that our approach can handle divergent motion patterns satisfactory well and shows an overall performance equivalent to the Horn and Schunck approach.

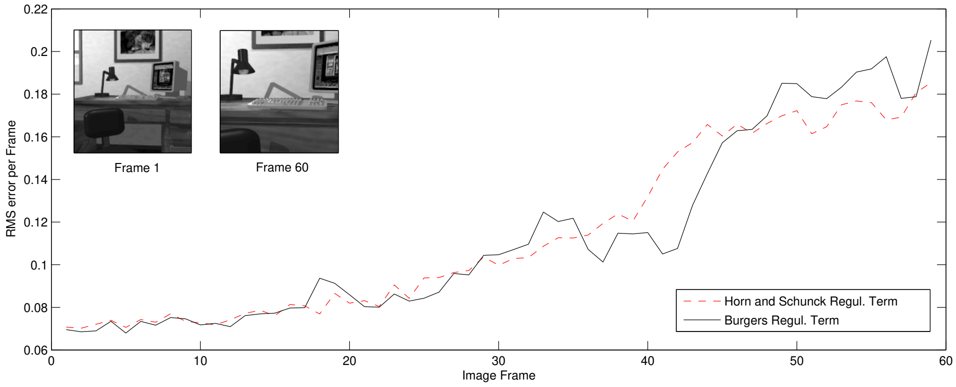


Figure 3.18: RMS error over the entire "office" sequence for the Horn and Schunck approach (2.8) and for our dynamic image motion approach (3.4). The first and last frame of the sequence are indicated in the corner of the figure.

Rotation: "Sphere" Sequence

The second motion pattern we intend to investigate is rotation. We constructed a simple image sequence containing a structured disk which rotates in front of a structured background. The results for the "rotating disc" sequence is shown on the left hand side of figure 3.19. We observe that our approach - due to the small motion - results in a rotational velocity field.

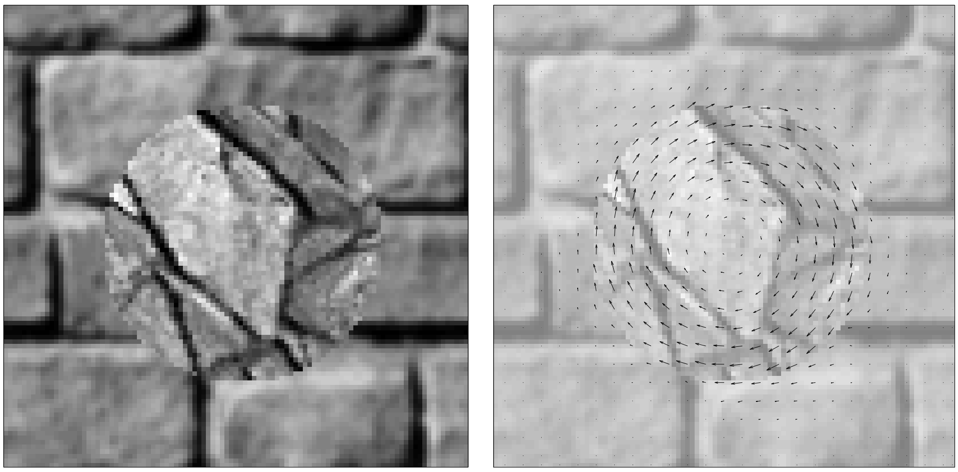


Figure 3.19: Synthetic image sequence: The "rotating disc" sequence. A structure disc rotates in front of a stationary background. **Left:** Frame 5 of the sequence. **Right:** Estimated optical flow field of the disc sequence computed with our approach (parameters: $\alpha = 0.01, \beta = 0.001$).

The "rotating sphere" sequence on which we intend to investigate our approach contains a curling vector field as well and is shown in figure 3.20. This sequence consists of 45 frames, where a sphere rotates (with a static motion field) in front of a stationary background.

The vector field we obtained by our approach (3.4) for this sequence is shown on the right side of figure 3.20. The shown vector field include an area within the white stripe near the boundary of the sphere that shows an interesting effect. The stripes in the background can easily confuse a motion estimation algorithm that is based on the optical flow assumption (2.2) as a motion field parallel to the stripes is grey value preserving (cf. aperture problem sec. 2.1.2). A small area of the background texture adopts to the motion that is present at the boundary of the sphere (perpendicular to the gradient of the image measurements I). As the chosen spatial regularisation parameter is relatively small for our approach ($\alpha = 0.008$) this area is not to large. With $\alpha = 0.01$ this area is more difficult for the Horn and Schunck approach (both parameters are optimised).

Inspecting the computed flow fields in figure 3.21 using the colour code shows this difference between our approach (3.21 bottom centre) and the Horn and Schunck approach (3.21 bottom right). While the latter one results in a smoother

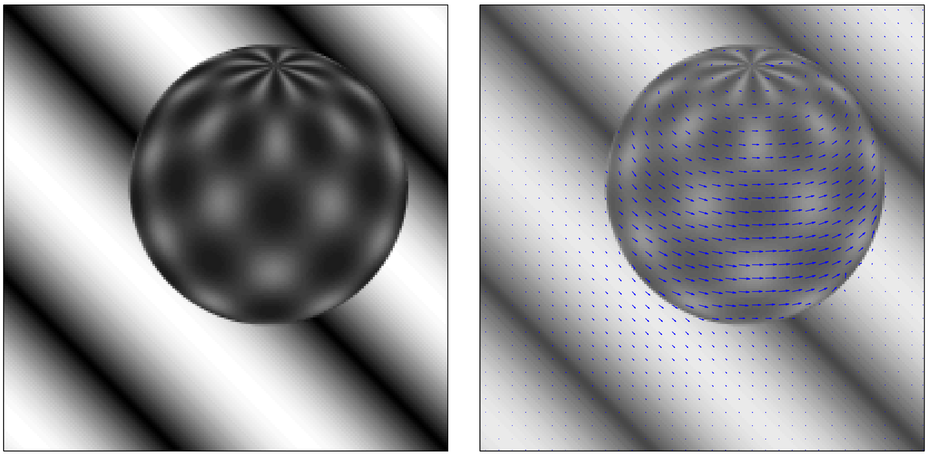


Figure 3.20: The synthetic "rotating sphere" sequence. The sphere rotates in front of a stationary background. **Left:** We depict frame 6 of the sequence. **Right:** Frame 7 along with the optical flow field obtained with approach (3.4) (parameters: $\alpha = 0.008$, $\beta = 0.0005$).

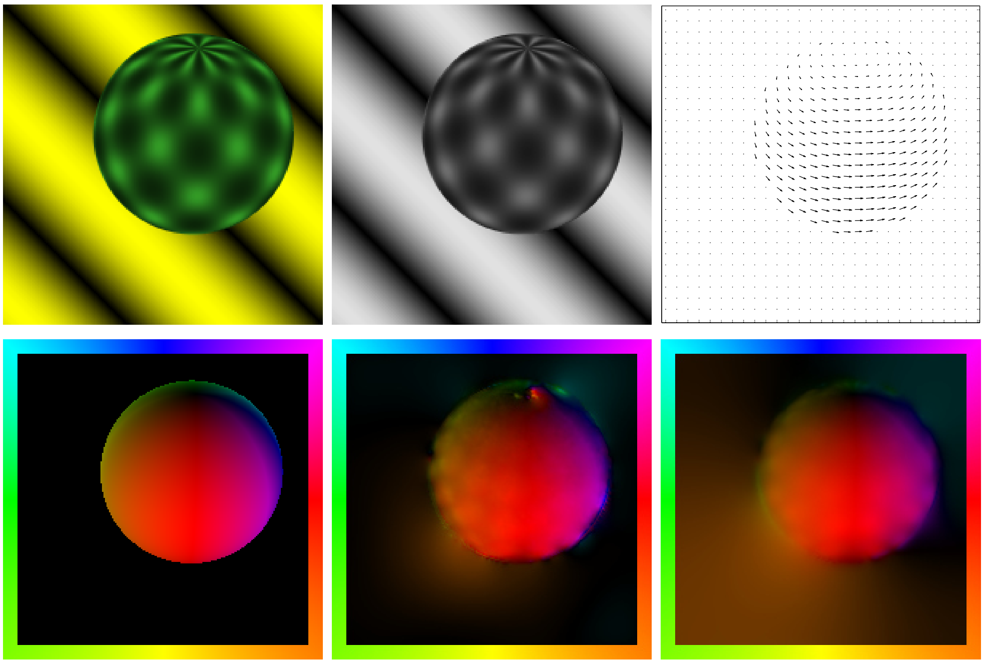


Figure 3.21: Qualitative results for a rotational motion pattern. **Top left:** Frame 6 of the "sphere" sequence. **Top centre:** Gray value version of frame 6 that is used in our computations. **Top right:** Vector plot of the ground truth data. **Bottom left:** Corresponding colour code representation of the ground truth vector data. **Bottom centre:** Optical flow field computed with our optical flow approach (3.4) for dynamic image motion (parameters: $\alpha = 0.008$, $\beta = 0.0005$). **Bottom right:** Optical flow field obtained by the Horn and Schunck approach (2.8) with homogeneous regularisation (parameter: $\alpha = 0.01$).

velocity field in the region of the sphere the motion estimation for the background is less accurate than for the Burgers approach.

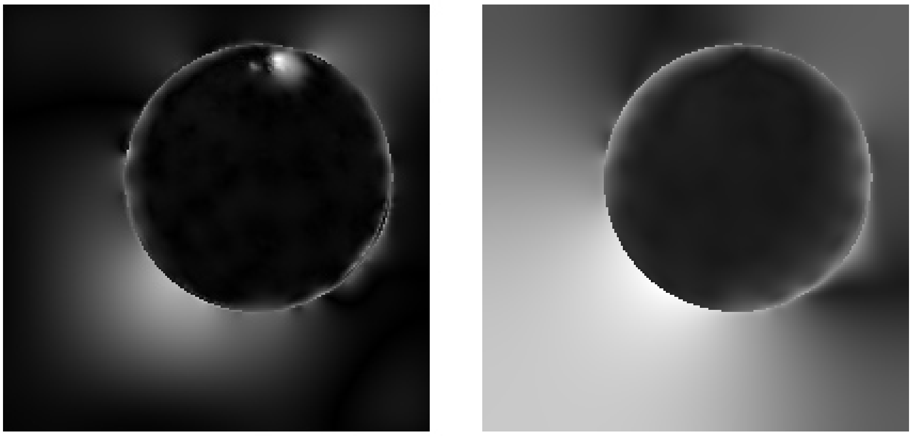


Figure 3.22: Error measure for the flow field obtained for the ”sphere” sequence. **Left:** AAE-plot for our result for frame 6 (AAE = 9.128°), (RMSE = 0.203). **Right:** AAE-plot for the Horn and Schunck result for frame 6 (AAE = 17.471°), (RMSE = 0.395). See text for further discussion.

For this sequence as shown in figure 3.23 the error measurements over the entire sequence reveals that our approach results steadily in a more accurate velocity field compared to the homogeneous regularised Horn and Schunck approach.

As discussed above the main error source for the Horn and Schunck approach is the smooth vector field that extends over the motion boundary into the background region. Our Burgers approach results in a more accurate flow field than the Horn and Schunck approach.

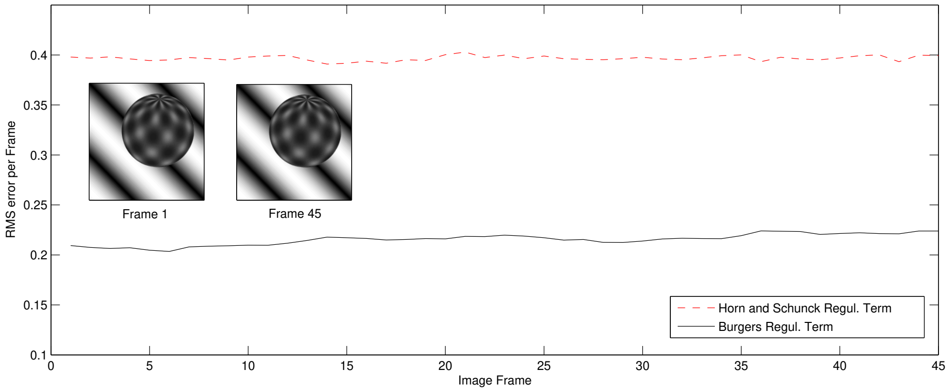


Figure 3.23: RMS error for the entire ”sphere” sequence. The comparison shows that the Horn and Schunck approach (2.8) results in a higher error than the approach for dynamic image motion (3.4). The top left corner shows the first and last frame of the zoom in sequence.

Combined Motion: "Street" Sequence

The next sequence we are concerned with is the "street" sequence [93] which covers a range of motion activity within 150 frames. That sequence shows two moving cars near a crossing while the camera slowly follows on of the cars. Figure 3.24 depicts the flow field we obtain with our approach for one of the later frames of the sequence. A camera slowly turns in the same direction the car is moving. This results in a more complex motion field with the background and the foreground moving into opposite directions.

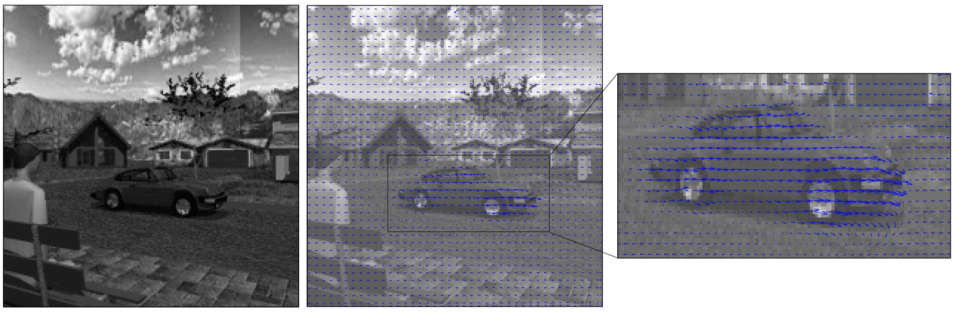


Figure 3.24: The "street" sequence, object and background are moving in opposite directions. **Left:** Frame 114 of the sequence. **Middle:** Frame 115 along with the estimated optical flow field obtained with our approach (parameters: $\alpha = 0.008$, $\beta = 0.0009$). **Right:** Details of the computed flow field.

The main difficulty of this image sequence lies in the fact that two motion fronts are moving against each other. The results for this interesting region which comprises the motion boundaries of the car is shown in detail in the right image of figure 3.24. This experiment provides us with some insight how motion boundaries are conserved by our approach and we evaluate this point in more detail below.

Figure 3.25 shows the results obtained by the Burgers approach (bottom row centre image of fig. 3.25) and the Horn and Schunck approach (bottom row right). Using also the error plots in figure 3.26 one can see that our approach leads to a more accurate motion boundary for the car. This is due to the fact that the regularisation term in our approach is a simplified mathematical model for shock flows enabling the Burgers approach to preserve the discontinuities in flow direction. Therefore we observe that the Burgers regularisation has an improving effect in the front area of the car leading to a more accurate motion field while the homogeneous smoothness regulariser from Horn and Schunck tends to have a smoother motion boundary there. Motion boundaries in computer vision represent discontinuities where a shock formation might help to model and preserve it.

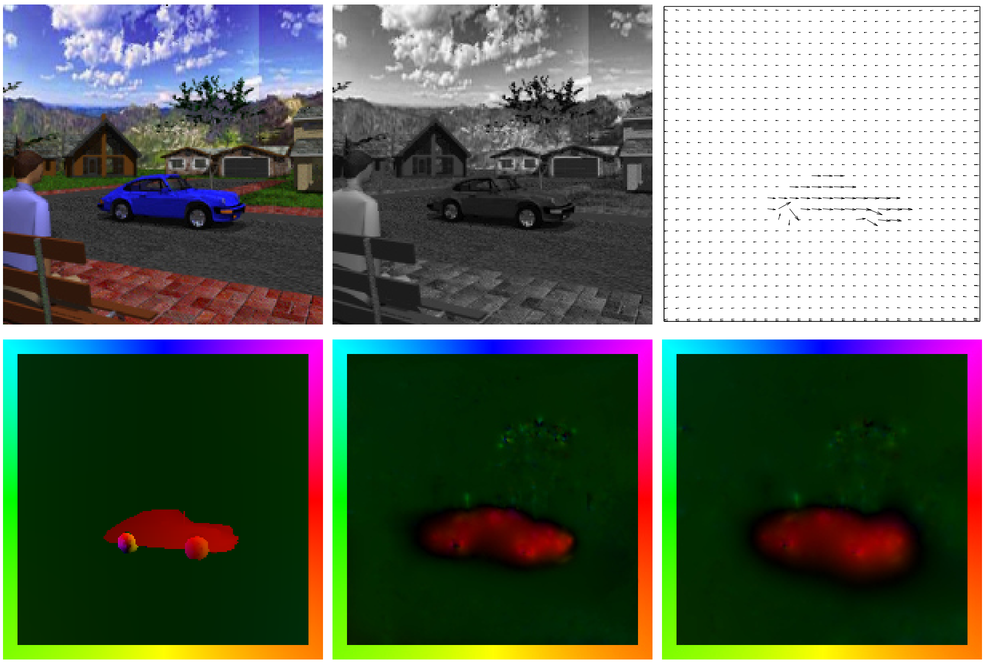


Figure 3.25: Qualitative results for the "street" sequence. **Top left:** Frame 114 of the "street" sequence. **Top centre:** Gray value version, respectively. **Top right:** Ground truth vector field for this frame. **Bottom left:** Colour code representation of the ground truth vector data. **Bottom centre:** Optical flow field computed with our dynamic image motion approach (3.4) (parameters: $\alpha = 0.008$, $\beta = 0.0009$). **Bottom right:** Optical flow field computed with the Horn and Schunck approach (2.8) with homogeneous regularisation (parameter: $\alpha = 0.008$).

The development of the RMS errors for the "street" sequence obtained by the Horn and Schunck approach (2.8) and the dynamic image motion approach (3.4) is provided in figure 3.27. We observe a similar performance for both approaches up to frame 80 after which the overall divergent motion increases and the second car comes nearer to the camera making a correct computation of the motion boundary more important. As the Burgers approach results in a sharper motion boundary in front of the car the total error for our approach is then smaller than the error for the Horn and Schunck approach.

From this image sequence we learned that the Burgers approach helps to improve the modelling of sharp motion boundaries for such kind of non-uniform motion pattern leading to an improved flow field in motion direction of the car.

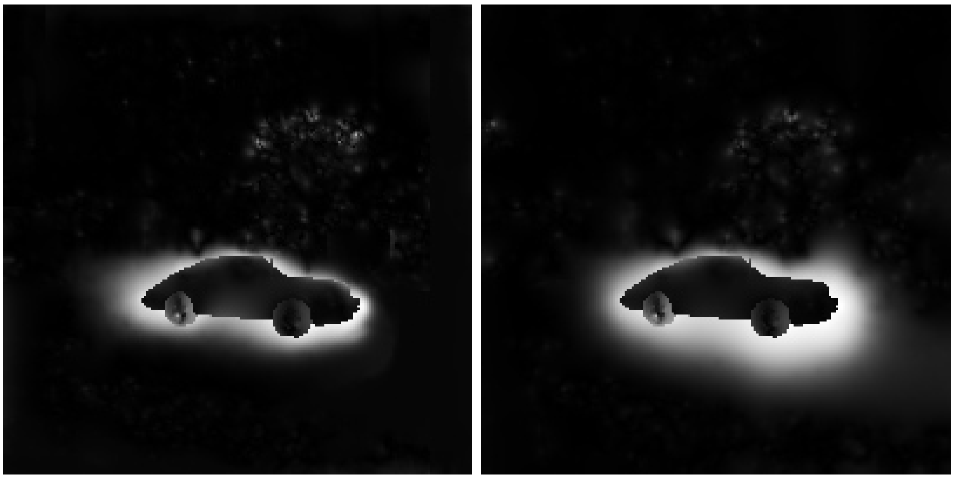


Figure 3.26: Comparison of the average angular error for the "street" sequence. **Left:** AAE-plot for frame 114 obtained with our approach (AAE = 7.746°, RMSE = 0.251). Due to the handling of shock formations the Burgers approach results in a better motion boundary in front of the car. **Left:** AAE-plot of the Horn and Schunck result for frame 114 (AAE = 9.878°, RMSE =0.290).

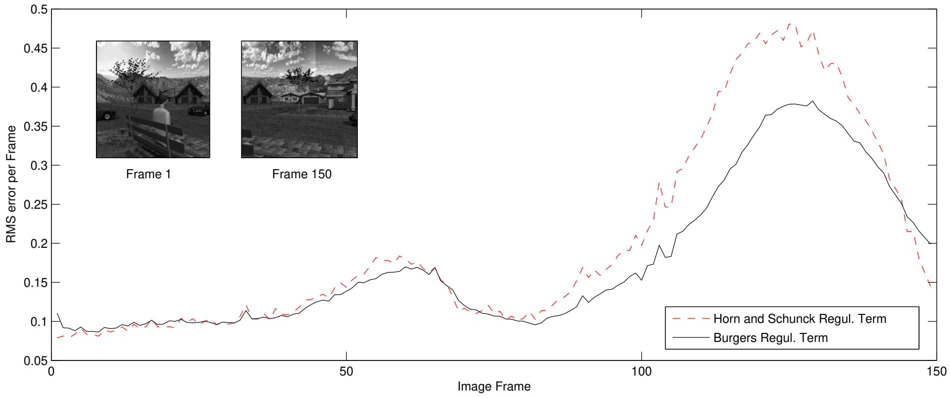


Figure 3.27: Comparison of the RMS error between the Horn and Schunck approach (2.8) and our dynamic image motion approach (3.4) for the entire "street" sequence (first and last frame are shown as thumbnails). The error plot shows a similar performance for the first half of the sequence while the Burgers approach outperforms within the latter part.

3.4.5 Noisy Image Data

The experiments in this section are intend to investigate the robustness of our approach under noise. We added random Gaussian noise with zero mean and increasing levels of the standard deviation $\sigma = 0,\ 5,\ 10,\ 20,\ 40$ to the "Yosemite" image sequence with clouds (available at `ftp://ftp.csd.uwo.ca/pub/vision`). The sequence exhibits divergent and translational motion combined with illumination changes.

The flow field obtained with our approach is shown exemplary in figure 3.28 for an added Gaussian noise with $\sigma = 40$. Note that we optimised the parameters according to the different noise levels. Generally, one can observe that with increasing noise level the smoothness parameter α has to be chosen higher to cope with the increasing perturbations (cf. table 3.1 for the parameters).

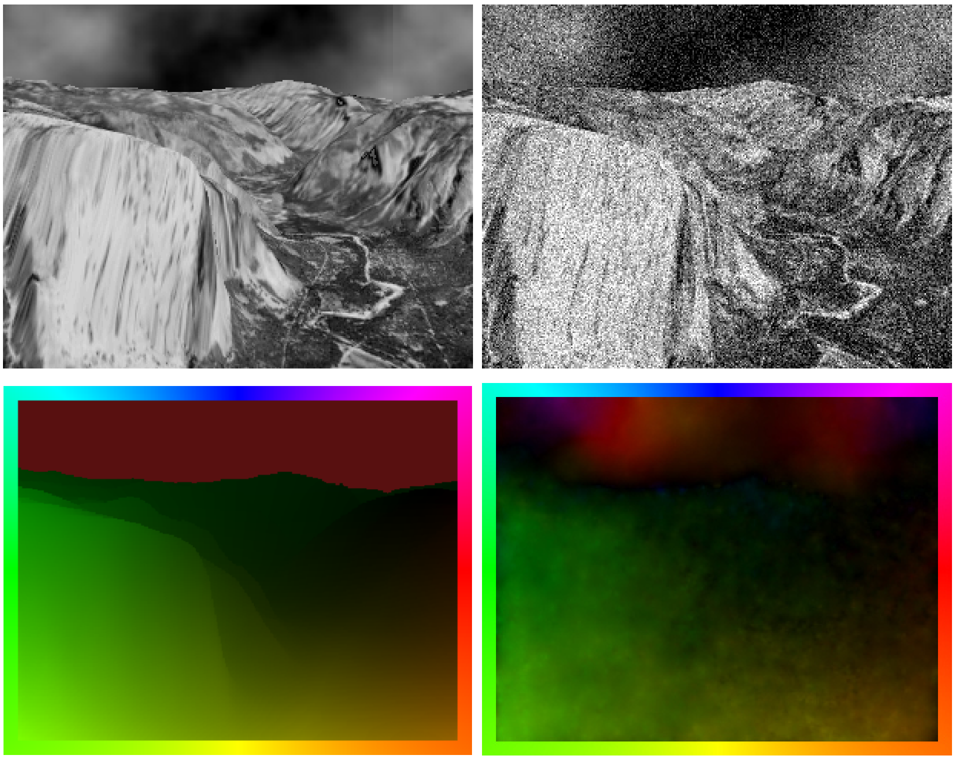


Figure 3.28: Robustness of our approach in presence of Gaussian noise. **Top Left:** Original image frame of the Yosemite sequence (with clouds). **Top Right:** We added Gaussian noise with standard deviation $\sigma = 40$. **Bottom Left:** Ground truth colour plot. **Bottom Right:** Computed flow field obtained by our dynamic image motion approach, (noise level: $\sigma = 40$, parameter: $\alpha = 0.05$, $\beta = 0.005$).

The particular results obtained by the approaches we compare at the different noise levels are summarised in table 3.1. In this table "DynMot" is the short cut for our dynamic image motion approach (3.4) and "HS" represents the approach of Horn and Schunck. We provide the RMS and the AA errors for different noise levels starting with no noise ($\sigma = 0$) and ending with a significant high noise level ($\sigma = 40$). Note that these are the mean values over the entire sequence. For a lower noise level we observe a similar performance for both approaches while at high noise levels our DynMot approach outperforms the HS approach. This behaviour can be understand as an effect that arises due to the temporal regularisation of the Burgers term. The smoothness parameter for the best results are quite similar for both approaches but β increased disproportionate for raising noise in our approach. This shows that with an increase of noise the perturbations can be better compensated by the temporal regularisation term namely the Burgers equation.

noise	approach	α	β	RMS	AAE
$\sigma = 0$	HS	0.005	-	0.1767	3.04°
	DynMot	0.006	0.00002	0.1784	3.09°
$\sigma = 5$	HS	0.005	-	0.2109	4.25°
	DynMot	0.006	0.00005	0.2090	4.22°
$\sigma = 10$	HS	0.008	-	0.2832	5.74°
	DynMot	0.01	0.0003	0.2750	5.68°
$\sigma = 20$	HS	0.02	-	0.4285	8.61°
	DynMot	0.025	0.001	0.3951	7.54°
$\sigma = 40$	HS	0.05	-	0.6402	13.27°
	DynMot	0.05	0.005	0.5226	9.89°

Table 3.1: Performance of our approach (DynMot) in comparison with the Horn Schunck approach (HS) in presence of noise: We added random Gaussian noise with zero mean and standard deviation σ to the "Yosemite" image sequence (with clouds). The parameters α and β are optimised for both approaches.

The results we obtained for our noise experiments reveal the following: The temporal regularisation of the new physical prior, the Burgers equation improves the capability of the variational approach to deal with significant noise levels.

3.4.6 A Computational Model of Motion Aftereffects?

Another interesting observation we would like to speculate about can be observed with the plot in figure 3.6. It shows the behaviour of the Burgers term for an abrupt stopping and starting event. The curve shows a fading characteristic which means that it decays after an event with a much lower rate indicating that it take some time to adapt to a new situation. In a rudimentary way, this is reminiscent of perceptual phenomena like motion aftereffects [91]. After adaption of the visual apparatus to some ongoing stimulus, a “negative” fading visual impression arises when the stimulus disappears. Of course, we are well aware that such an interpretation is highly speculative.

3.4.7 Temporal Regularisation

In figure 3.30 we compare the RMS error of the flow obtained for three different approaches. In particular these are the Horn and Schunck approach (2.8), the image-driven variational approach (2.15), and our approach (3.4). Note that (2.15) omits the influence of the Burgers equation (3.1) in (3.4). The image sequence we are concerned with is depicted in figure 3.29.

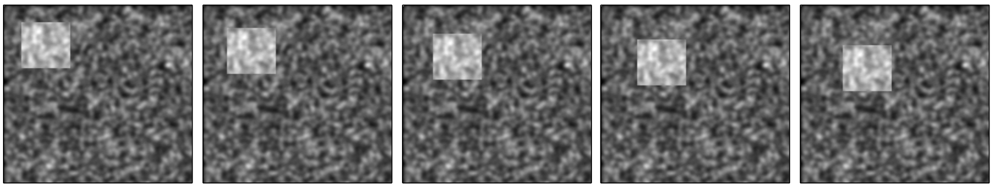


Figure 3.29: Synthetic image sequence: The image sequence consists of 21 frames and shows the movement of a 40x40 pixel patch, which moves diagonal in front of a textured background $5\sqrt{2}$ pixels per frame from the upper left corner to the down right corner.

The sequence consists of 20 frames showing a textured patch (40x40 pixels), which moves diagonal in front of a textured background from the upper left corner to the lower right corner with a speed of $5 * \sqrt{2}$ per frame. The parameters have been optimised.

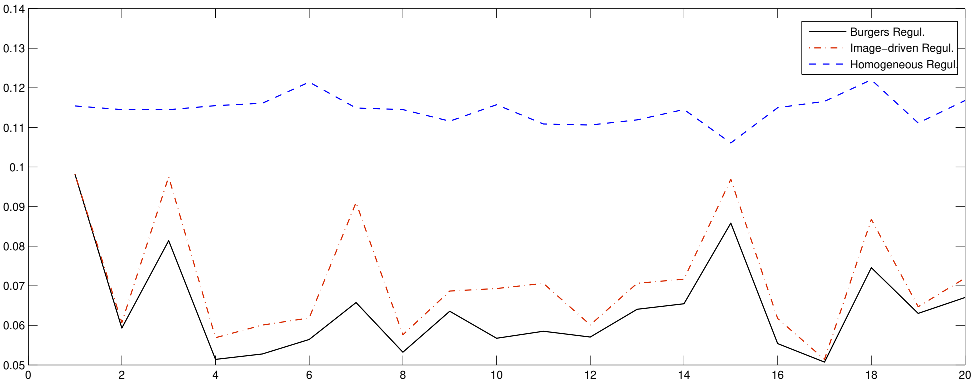


Figure 3.30: Comparison of the RMS error between the purely spatial homogeneous regularisation approach from Horn-Schunck (2.8), the image-driven variational approach (2.15), and the dynamic approach (3.4). The latter combination allows for temporal regularisation with a recursive computational architecture, leading to consistently lower estimation errors in particular if the actual and expected motions agree (parameters: $\alpha = 0.01$, $\beta = 0.001$).

The homogeneous regularised Horn and Schunck approach [67] results in the highest RMS error. This approach is improved by the image-driven approach of Nagel (2.15) which attains a better optical flow estimation. The dynamic approach using physical prior knowledge shows the lowest RMS error as the physical prior knowledge resulting in an additional temporal coherency, which makes the approach less independent from the actual image data. Recall that the temporal coherency results from the transport mechanism of the Burgers equation. The consistently lower error indicates that the ability of motion prediction is an advantage for the estimation of the motion and exerts temporal regularisation.

3.5 Conclusion

We presented a variational approach to image motion estimation incorporating the Burgers equation as physical prior knowledge. In order to regularise the optical flow constraint equation, we added a matching term that penalises the deviation from predicted velocity values. While this regularisation term penalises deviations between the current velocity and the propagated velocity of the preceding frame, it does not enforce spatial smoothness of the velocity, therefore, we added the spatial image-driven regularisation term of Nagel.

The resulting distributed-parameter approach performs spatio-temporal regularisation in a recursive online fashion in the sense that as soon as the next image is obtained the results for the previous image frame is used to compute the current motion estimation. This recursively involves the flow field that is computed using previous results and means that the computation can be done in a sliding window manner where velocities can be computed as soon as their respective frames have been recorded.

We experimentally explored the properties and potential benefits of the presented variational approach. The most prominent features that our approach shows is its inertia behaviour along with its temporal regularisation. We found that our approach - due to the constant velocity assumptions of the Burgers equation - predicts the motion of uniform motion patterns quite well and less accurate for non-uniform motion patterns. However to a certain degree it is also useful for non-uniform motion and we observed in both cases that the estimated motion field adapts - after recognising a sudden change of motion - nicely to the new image measurements. We demonstrated that deviations from the expected velocity distribution generate vector fields that may serve as attentional mechanism for a superordinate processing stage and it is tempting to point out a potential relevance of our approach for related models of visual perception. However, as we are not experts in this field, we confined ourselves to a few speculative remarks. Our occlusion experiments showed, that the resulting model exhibits, properties analogous to a dynamic filter for the most elementary polynomial kinetically model of a point feature (cf. [6]): in the absence of image measurements, and if spatial context does not indicate otherwise, velocities do not change. In this way, according to the underlying constant-velocity assumption of the Burgers equation, the formulation may be regarded as an elementary distributed-parameter version of a low-order polynomial filter commonly used for the kinematics of point features. The effect of temporal regularisation is well observable on image sequences with a very high noise level where compared to the Horn and Schunck approach the temporal smoothness lead to improved motion estimations. Another interesting feature is that the shock front of a transported motion field resembles a sharp motion boundary which is often desired in optical flow computations. However, this effect occurs only in motion direction. Beside these interesting new properties that are inherent in our approach we experimentally observed a similar performance as the Horn and Schunck approach. Here we focused on the exploration of the newly shown properties.

To summarise, our dynamic image motion approach combines variational motion estimation with motion prediction through a transport process based on the Burgers equation as new physical prior exploiting the knowledge that moving structures should exhibit some inertia, meaning that velocities are not expected to change.

4

Control Based Optical Flow Estimation

In the previous chapter we presented a variational approach to dynamic image motion computation using a physical motivated regularisation term. This new matching term represents the difference of the expected and apparent motion field with respect to the solution of the Burgers equation. This difference field can be interpreted as a force that adapts the expected motion towards a motion field consistent with the image measurements. In this chapter we follow the idea of the control based interpretation and propose directly an optimal control approach for optical flow estimation. It is an attempt to adopt control methods from the field of applied mathematics in a new form to image sequence processing and to provide a first evaluation of the capability of this approach. The application of optimal control techniques to image motion estimation, as presented in this work is novel and explorative.

4.1 Optimal Control Formulation

In this chapter we study a novel optical flow-based approach to image motion analysis, that incorporates physical prior knowledge in a control framework. Our motivation draws on literature on the control of distributed parameter systems in connection with fluid dynamics [58]. We are aware of the works of [116, 16], where the authors present a control approach for optical flow estimation. We interpret the grey values of the images as a ”fictive fluid” and incorporate the Burgers equation, in order to model the physical fluid motion behaviour of the image sequences. As a consequence, flow fields are forced to vector fields which should satisfy the Burgers equation. In order to estimate the motion of apparent velocities of image measurements in an image sequence, control variables are included and determined by minimising an appropriate objective functional. Control variables and optical flow adjust to the observed image data.

We adopt the variational optical flow approach of Horn and Schunck (2.8) and add a distributed control resulting in a constrained minimisation problem. The obtained objective functional has to be minimised with respect to the optical flow and control variables subject to the Burgers equation over the entire flow domain in space and time. Therefore, our approach estimates not only the optical flow data from an image sequence, but it also estimates a force driven by the Burgers equation. The force field indicates the violation of the Burgers equation and acts in motion situations where the Burgers equation is not the right model to describe the motion - for example accelerated motions as starting or stopping events - or even the change of the motion direction. This is similar to the previous chapter where we proposed to use the difference field between the "seen" and the "expected" motion as an attentional mechanisms for unexpected motion events. However, in the approach we propose here the force/control - indicating the deviation of the velocity field from the Burgers behaviour - is penalised directly in the objective functional.

For the solution of the control based optical flow estimation problem we employ the Lagrange multiplier method providing us with a tool to find extrema of functionals subject to specific constraints. The initially constrained optimisation problem is reformulated into an unconstrained problem allowing to obtain the associated first-order optimality system [58]. This results in a forward-backward system with appropriate initial and boundary conditions, meaning that the optimality system consists of a backward-in-time equation where a terminal condition at $t = T$ is given and a forward-in-time equation with an initial condition at $t = 0$. The optimality system is a coupled system and can be solved in one shot, i.e., without decoupling the system. However, this results in a very large system, where the solutions at all time levels are coupled to each other leading to an expensive computation. Another way - which we follow - to obtain the solution is to uncouple the forward and backward computation as described in [58] leading to an iterative solution scheme.

While our previous approach results in a recursive online computation, here we have to solve the system - like the approach of Weickert and Schnörr (2.16) - in a batch processing mode as a spatial-temporal regularisation over the entire integration domain. In order to obtain a sliding window scheme we have to employ an appropriate receding horizon control approach. There the current control and state are determined by minimising the energy functional over a subtime interval $[t, t + T]$. This architecture allows to obtain - like within the previous approach - a velocity estimate that can be regarded as a *predicted* vector field for the next time interval. The computation can be done in a semi non-batch processing mode. However, we address this extension to a further work project.

4.1.1 Overview

The remaining parts of this chapter are organised as follows. First we summarise two approaches that can be found in the literature which illustrate the feasible combination of optical flow approaches and fluid control methods. In section 4.2 we present our control approach for motion estimation which incorporates the Burgers equation as physical prior. The derivation and solution of the optimality system is subject of section 4.2.1 and 4.2.2. An overview of our implementation is given in section 4.2.3. In order to test our algorithm, we perform numerical experiments in section 4.4 on synthetic and real image sequences. The results demonstrate that the optimal control formulation has a similar performance as our previous approach. However, due to the global regularisation architecture of the approach, the temporal regularisation tends to be better for highly noisy image sequences.

4.1.2 Relevant Control Approaches for Optical Flow

In this section we present the ideas of two existing control approaches that are related to motion computation of image sequences.

Ruhnau and Schnörr presented in [116] an optical flow estimation approach for particle image velocimetry that is based on a control formulation subject to physical constraints. Their aim is to estimate the velocities of particles in image sequences of fluids. They stated the following energy function

$$E(u,p,f,g) = \int\limits_{\Omega} \frac{1}{2}(\partial_t I + u \cdot \nabla I)^2 \, dx + \int\limits_{\Omega \setminus \Omega_0} \alpha \frac{1}{2}|f|^2 dx + \int\limits_{\Gamma} \gamma \frac{1}{2}|\nabla_\Gamma g|^2 d\Gamma,$$

which has to be minimised subject to the *time-independent* Stokes system

$$\begin{aligned} -\mu\Delta u + \nabla p &= f \quad \text{in } \Omega, \\ \nabla \cdot u &= 0 \quad \text{in } \Omega, \\ u &= g \quad \text{on } \Gamma, \end{aligned}$$

where u represents the velocity field, μ the viscosity, p the pressure and f the body force acting on the fluid. The authors denote regions where large forces are expected by Ω_0 and exclude the body force penalisation at these locations. The authors exploit also the optical flow constraint equation (2.2) in their objective functional but rather than to penalise the estimated flow directly they regularise the body force f and derivatives of the values g on the boundary and minimise the functional subject to the Stokes system on the spatial image domain Ω. The appropriate optimality system is then solved resulting in optimal states and controls. This control approach forces the velocity field to hold the time-independent Stokes system as good as possible and relies on the measurement of two consecutive image frames. Our approach which has to satisfy the time-dependent Burgers equation leads to a forward-backward computation of *time-dependent* equations.

Another control approach to image motion estimation has been proposed by Borzi, Ito and Kunisch in [16]. The basic idea of their approach is to estimate both an optical flow field u and a rectified image function I satisfying the flow constraint equation (2.2). Note that in their approach Y_k (and not I) denotes the sampled images of the image sequence and their optimal control formulation consists in minimising the following objective function:

$$\begin{aligned} J(I,u) &= \frac{1}{2}\int_{\Omega}\sum_{k=1}^{N}|I(x_1,x_2,t_k)-Y_k|^2dx \qquad (4.1)\\ &+ \frac{1}{2}\int_{\Omega\times(0,T]}\left\{\alpha\Phi\left(|\partial_t u|^2\right)+\beta\Psi\left(|\nabla u_1|^2+|\nabla u_2|^2\right)+\gamma|\nabla\cdot u|^2\right\}dxdt \end{aligned}$$

with respect to

$$\begin{aligned} \partial_t I + u\cdot\nabla I &= 0 \quad \text{in } \Omega\times(0,T], \qquad (4.2)\\ I(\cdot,0) &= Y_1. \end{aligned}$$

The most significant difference to our optical flow approach is that they do not only estimate the optical flow u, but also I_k which is an approximation of the captured grey value distributions Y_k, where k specifies the frame number within the image sequence. As part of the first-order necessary optimality conditions of the Lagrangian functional their optimal control formulation does not require a differentiation of the image data. The control problem (4.1-4.2) results in the problem of finding the solution of two forward-backward hyperbolic equations and two elliptic equations.

4.2 Estimation Approach

In this section we explain our optical flow control approach in more detail. Our control approach draws on the literature of optimal control approaches in connection with fluid dynamics. Several methods which we exploit in our approach can be found in the book of Gunzburger [58].

In order to regularise the optical flow constraint (2.2) standard variational approaches like (2.8) or (2.16) add a constraint to enforce spatial or spatio-temporal smoothness of the optical flow field, respectively. In this chapter we investigate a control based formulation of our optical flow estimation approach. We obtain this spatial-temporal approach in a control based framework as follows: We introduce a control f, that is distributed in space and time, which means that it acts over the entire optical flow domain $\Omega\times[0,T]$. The amount of the control is regulated through the penalisation within the objective functional

$$\begin{aligned} J(u,f) &= \frac{1}{2} \int\limits_{\Omega\times[0,T]} \Big\{ (\partial_t I + (u\cdot\nabla)I)^2 \\ &\qquad\qquad + \alpha(|\nabla u_1|^2 + |\nabla u_2|^2) + \beta|f|^2 \Big\}\, dxdt, \end{aligned} \tag{4.3}$$

subject to the following physical constraint equation

$$\left\{ \begin{array}{rcll} \partial_t u + (u\cdot\nabla)u &=& f & \text{in } [0,T]\times\Omega, \\ \partial_n u &=& 0 & \text{on } [0,T]\times\Gamma. \end{array} \right. \tag{4.4}$$

The left hand side of the first equation of (4.4) is the *material derivative* of the velocity field u and is known as the Burgers equation (cf. sec. 7.5). It describes a moving fluid where the positions of fluid particles change with time while a (fictive) *body force* $f = (f_1, f_2)$ which represents the control is acting on the fluid. In (4.3) the spatial smoothness of the vector field u is incorporated by the α-term and the parameter β penalises the occurrence of the force f. As the β-term in (4.3) should be as small as possible, the flow field u is forced to hold the Burgers equation as good as possible. Therefore, for higher β values one expects smaller control values f.

The main objective of our control formulation is to determine a body force f (the control !) along with the velocity field u which should match the apparent motion measured - by the means of the optical flow constraint equation - as well as possible. The magnitude of the control is limited through the penalisation within the objective functional.

In terms of control theory (e.g.,[58]) the approach can be described as follows: We want to find an optimal state $u = (u_1, u_2)^\top$ along with an optimal control $f = (f_1, f_2)^\top$, such that the functional $J(u, f)$ is minimised and u and f satisfy the Burgers state system (4.4).

In contrast to the approach (3.4), where we introduced a matching term penalising the derivation between the current velocity and the propagated velocity of the preceding frame, we employ the body force directly satisfying the Burgers equation over the entire image domain $[0,T]\times\Omega$. However, we expect a similar behaviour of both approaches. The minimisation of (4.3) should result in a velocity field which matches the apparent motion as well as possible, while being pushed towards a Burgers transport by as little force as possible (cf. experiment sec. and fig. 4.14).

4.2.1 Optimality System

To derive the optimality system for the constrained optimisation problem (4.3 - 4.4) we recast it into an unconstrained optimisation problem. Introducing the Lagrange multiplier $w = w(x,t)$ we obtain the Lagrangian functional

$$L(u, f, w) \quad = \quad J(u, f) - \int\limits_{\Omega\times[0,T]} w \cdot (\partial_t u + (u \cdot \nabla)u - f)\, dxdt, \qquad (4.5)$$

where the multiplier $w = (w_1, w_2)^\top$ is also known as the corresponding adjoint variable to the state variable u in (4.3 - 4.4).

The first-order necessary conditions [58] for an optimum of (4.5) result in the following optimality system from which the optimal state u and the optimal control f can be determined:

$$\textit{state eq.} \quad \begin{cases} \partial_t u + (u\cdot\nabla)u &= f & \text{in } \Omega\times[0,T], \\ \partial_n u &= 0 & \text{on } \Gamma\times[0,T], \\ u|_{t=0} &= u_0 & \text{in } \Omega, \end{cases} \qquad (4.7a)$$

$$\textit{adjoint eq.} \quad \begin{cases} -\partial_t w - (u\cdot\nabla)w - w\nabla\cdot u + (\nabla U)^\top w & \\ \quad = \nabla I(\partial_t I + u\cdot\nabla I) - \alpha\Delta u & \text{in } \Omega\times[0,T], \\ \qquad w = 0 & \text{on } \Gamma\times[0,T], \\ \qquad w|_{t=T} = 0 & \text{in } \Omega, \end{cases} \qquad (4.7b)$$

$$\textit{optimality cond.} \quad \begin{cases} \beta f + w &= 0 & \text{in } \Omega\times[0,T], \\ f &= 0 & \text{on } \Gamma\times[0,T], \\ f|_{t=T} &= 0 & \text{in } \Omega, \end{cases} \qquad (4.7c)$$

where

$$(\nabla U) = \begin{pmatrix} \partial_{x_1} u_1 & \partial_{x_2} u_1 \\ \partial_{x_1} u_2 & \partial_{x_2} u_2 \end{pmatrix}$$

denotes the Jacobian matrix of u [32]. The state equation (4.7a) is obtained by derivating the Lagrangian functional (4.5) in the direction of the Lagrange multiplier, and turned out to be identical to the Burgers system equation (4.4) itself. The adjoint equation (4.7b) specifies the first-order necessary conditions with respect to the state variables u. The optimality condition (4.7c) is the necessary condition that the gradient of the objective function – with respect to the control f – vanishes at the optimum. The detailed derivation of the optimality system (4.7a - 4.7c) is provided in appendix A.1. In the next section we describe how the optimisation problem can be solved.

4.2.2 Solution of the Optimality System

The optimality system (4.7a - 4.7c) is a coupled system which turns out to be - due to the large number of unknowns - prohibitively expensive to be solved directly. To solve this system we introduce an iterative method which decouples the state and adjoint computation. This results in a gradient descent method which consists of the iterative solution of the state and adjoint equation in such a way that the state equation is computed forward in time with appropriate initial condition u_0 and the adjoint equation is computed backward in time with terminal condition $w_{t=T} = 0$. The optimality condition is used to update the control f with the adjoint variable w. The control f is then used to compute the actual state u. Then the step length is adjusted ensuring that the actual energy of the objective functional (4.3) is smaller than in the previous iteration. Note that we choose the start value for f to be zero in the very first iteration.

Gradient of the Objective Functional

To determine the optimal state and control we use - as mentioned above - a gradient descent method. Therefore, we have to determine the gradient of our objective functional (4.3) with respect to the control f. The state equation (4.7a) is solved to determine the state u as a function of the control f so that the functional $J(u, f) = J(u(f), f)$ is a function of only the control f. Applying the chain rule to $J(u(f), f)$ we obtain the total derivative of the functional (4.3):

$$\partial_f J = \frac{\partial J}{\partial u}\frac{du}{df} + \frac{\partial J}{\partial f}. \tag{4.8}$$

However, the full derivation of the gradient of the objective functional (4.3) is presented in detail in appendix A.2. In the following we will only provide the main points of the deviation. We specify the gradient $\frac{du}{df}$. This term defines the so-called *sensitivities* and describes the change of the state when the control variable changed. To obtain this dependency we can use a variation of the distributed control $f + \varepsilon\tilde{f}$ which is assumed to correspond to the changes of the state u to $u + \varepsilon u_f$. The change u_f is determined by the state system:

$$\begin{cases} \partial_t(u + \varepsilon u_f) + ((u + \varepsilon u_f) \cdot \nabla)(u + \varepsilon u_f) &= f + \varepsilon\tilde{f} \quad \text{in } \Omega \times [0, T], \\ \partial_n(u + \varepsilon u_f) &= 0 \quad \text{on } \Gamma \times [0, T]. \end{cases}$$

As derived in A.2 one obtains for $\varepsilon \to 0$ the sensitivity equation

$$\begin{cases} \partial_t u_f + (u_f \cdot \nabla)u + (u \cdot \nabla)u_f &= \tilde{f} \quad \text{in } \Omega \times [0, T], \\ \partial_n u_f &= 0 \quad \text{on } \Gamma \times [0, T], \end{cases} \tag{4.9}$$

which describes the fact that an infinitesimal variation of the control in the direction of the control induces an infinitesimal variation in the direction of the local velocity u_f.

The change in the functional $J(u, f)$ of (4.3), effected by an infinitesimal change in the direction $\tilde{f}$ in the control f leads us to the gradient of the objective functional (we refer for details to sec. A.2):

$$\partial_f J = w + \beta f. \tag{4.10}$$

This gradient of our objective functional is used within our iterative gradient descent algorithm to update the search direction. Below we sketch the implementation of the gradient method.

4.2.3 Algorithm

In this section we describe the algorithm for the gradient descent method which includes an automatic step-length selection to find a minimum of the optimal control system ($4.7a$ - $4.7c$).

At the beginning of the algorithm the initial flow fields for all image frames are set to zero ($u = 0$). Starting with a terminal condition of the adjoint variable for the last frame $w_{t=T} = 0$ the adjoint equation is solved backward in time resulting in adjoint variables w for all image frames. The optimality condition is then used to update the control f for all frames. With this the state equation is used to compute u, i.e. the values for all flow fields forward in time. Depending on the objective value of J the step length is adapted. If the convergence criteria is not yet reached the next iteration starts with solving the adjoint equation again.

In the following pseudo code description of the algorithm, s denotes the step-size parameter that is adapted by the algorithm. The parameter ϵ denotes the threshold which is used to decide whether the relative difference of the energy is small enough to be seen as converged.

Algorithm 1 Gradient algorithm with automatic step-length selection

```
 1: set all flow fields initially to $u = 0$
 2: set all control fields initially to $f = 0$
 3: choose tolerance $\epsilon$
 4: $s := s_0$ (initial step size)
 5: repeat
 6:     solve the adjoint equation ($4.7b$) for $w$
 7:     update $f$: $f_m = f_{m-1} - s(\beta f_{m-1} + w)$ with ($4.7c$)
 8:     solve the state equation ($4.7a$) for $u$
 9:     if $J(u, f_m) \geq J(u, f_{m-1})$ then
10:         $s := 0.5s$
11:         GOTO 7
12:     else
13:         $s := 1.5s$
14:     end if
15: until $|J(u, f_m) - J(u, f_{m-1})| / |J(u, f_m)| < \epsilon$
```

As one has to solve the optimality system ($4.7a$ - $4.7c$) within the algorithm we describe our implementation of the equations in the next section.

4.3 Numerical Solution

As ($4.7c$) can be used directly to update the actual control f we are here concerned how the non-linear state equation ($4.7a$) and the linear adjoint equation ($4.7b$) can be discretised and numerically solved. Generally, the state equation is solved forward in time over the entire image sequence with a given start value u_0 at time $t = 0$. The adjoint equation is solved backward in time with a given terminal value w_T at time $t = T$. In our implementation we approximate the initial optical flow field of the image sequence by $u_0 = 0$ and $w_T = 0$.

In the next subsections we are concerned with the discretisation of the state and adjoint equation respectively. However, we refer in each section to the appropriate numerical part of this thesis for a more detailed description.

4.3.1 Discretisation of the State Equation

The numerical implementation of the non-linear state system equation $(4.7a)$ is quite difficult and we employed a second-order conservative Godunov Method as described in section 8.3.2. We have to cope with over- and undershoots, with several types of shock formations, with the compliance of conditions (entropy-, monotony-, CFL-condition, etc.) and some different discretisation schemes (cf. sec. 8.3). The fluxes are numerically computed by solving the equations at pixel edges. The reasonable behaviour at discontinuities is obtained by using the solutions of the appropriate Riemann problem (8.47) incorporating Van Leer limiters (cf. sec. 8.2.1) to prevent over and undershoots. For the full discretisation procedure of the state equation $(4.7a)$ we refer to section 8.3.

4.3.2 Discretisation of the Adjoint Equation

In order to solve the time-dependent adjoint equation $(4.7b)$ we employed a conservative finite difference method, more specifically Fromm's second-order upwind method (see sec. 8.2.2). The numerical fluxes are obtained by solving the characteristic form at the cell edges, while the fluxes are evaluated and differenced at the same time. The basic idea is to satisfy Godunov's theorem in a naturally way. Roughly speaking, Godunov's theorem says that all methods of accuracy higher than order one will produce spurious oscillations in the vicinity of large gradients, while being second-order accurate in regions where the solution is smooth. Therefore the slopes of Fromm's second-order method (8.2.2) are in regions with a large gradient replaced by slops of Van Leers scheme (cf. sec. 8.2.1), which detects discontinuities and modifies its behaviour accordingly. The implication of this hybridisation is that the resulting method retains the high-order accuracy of Fromm's method in smooth regions, but where discontinuities are detected, the discetised evolution equation drops to first-order accuracy. The discretisation of the adjoint equation is described in more detail in section 8.3.3.

4.4 Experiments

In this section we assess our control approach by applying it exemplarily to synthetical and real image sequences (as depicted in fig. 4.1) and we provide a comparison with results obtained by our previously developed optical flow approach.

Figure 4.1: "Waving hand" image sequence: Five consecutive images of our real image sequence.

In particular, this section is divided into the following parts:

- First, we show exemplarily the numerical behaviour of the energy minimisation process.
- Secondly, we discuss the parameter selection for the control approach along with discussing the influence of the Burgers equation as control.
- As third task, we illustrate the behaviour of our optical flow control approach on a real-world 2D image sequence with respect to unexpected motion events.
- Then we present the results for noisy image data showing the influence of the temporal regularisation in the control approach and provide a comparison with error measures obtained by the dynamic optical flow approach.
- Finally, we show results for a real-world 2D image sequence in comparison with the results obtained by a spatio-temporal approach.

4.4.1 Numerical Convergence

In this section we provide a typical example of the numerical convergence of the energy minimisation process over 5000 iterations (fig. 4.2). We demonstrate the effect of the energy minimisation at some intermediary computational results at iteration 10, 100, 1000 and 4000 (fig. 4.3). The energy plot is shown in figure 4.2. Note that the plot has a logarithmic scale in order to provide a better impression of the evolution.

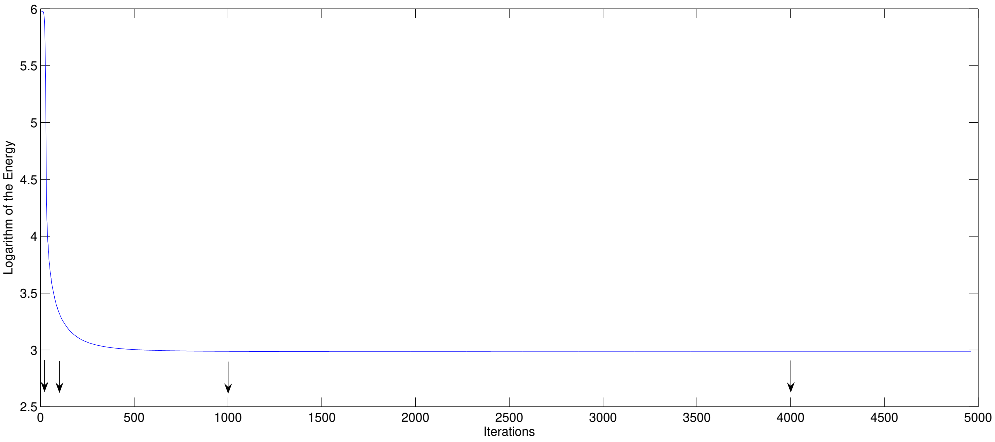


Figure 4.2: Energy evolution for the "waving hand" sequence over 5000 iterations (the logarithm of the measured energy values is shown). With initial approximation $u_0 = 0$ and $w_T = 0$. The arrows indicate the iterations for which the appropriate vector fields are shown in figure 4.3. After approximately 500 iterations it is reasonable to stop the minimisation process.

In our numerical implementation we use an initial optical flow field $u_0 = 0$ and for the terminal adjoint variable $w_T = 0$. One can observe that the energy measure is converged after about 500 iterations. This indicates that it is reasonable to terminate similar computations after about 500 iterations.

However, to illustrate the evolution of the flow field during the minimisation we choose four investigation points at iteration 10, 100, 1000 and 4000. For these points we depict the appropriate results of the optical flow computation in figure 4.3. The Euclidean distances between the flow fields of two consecutive investigation points are shown as well. As expected from the evolution of the energy minimisation we observe the largest changes at the beginning of the iteration process. A more detailed investigation of the Euclidean distance measure reveals that after iteration 1000 only very small corrections occur that led to a slightly smoother flow field estimation (cf. fig. 4.4).

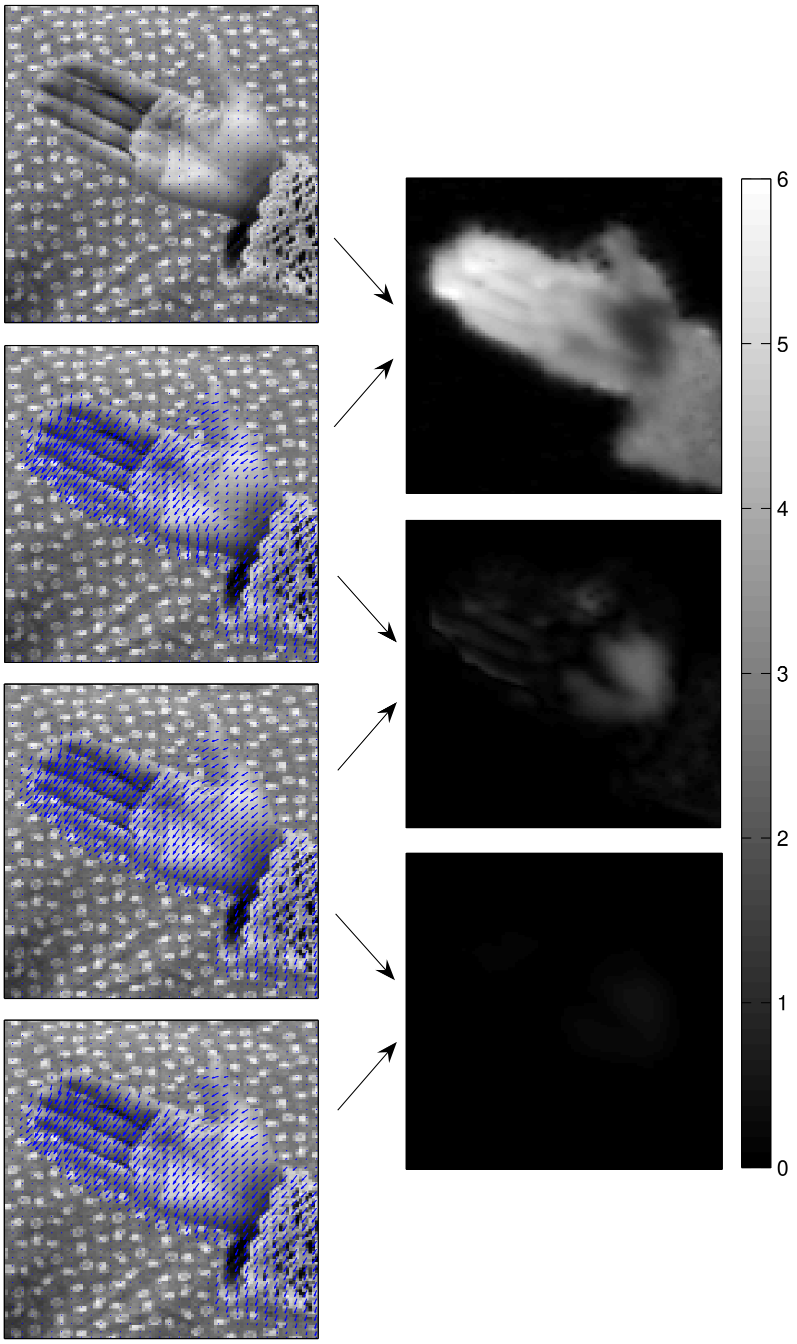


Figure 4.3: Computed optical flow fields at iteration 10, 100, 1000 and 4000 (left column). The right column shows the Euclidean distance between the optical flow field of iteration 10 and 100, 100 and 1000, and 1000 and 4000 and indicates the changes over the iterations. In accordance with the energy minimisation process we observe that the largest change occur at the beginning of the iteration process. In order to reveal what happens after iteration 1000 we illustrate the Euclidean distance measure for the last flow field pair with a different scaling in figure 4.4.

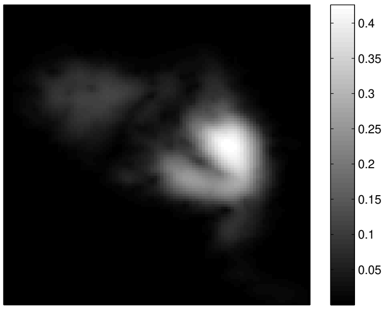


Figure 4.4: The Euclidean distance measure between the optical flow field of iteration 1000 and 4000 with a new scaling shows the difference of the flow field more clearly. The changes after iteration 1000 have only a small contribution resulting in a slightly smoother optical flow field.

4.4.2 Parameter Selection

To provide an idea of the influence of the control parameter we depict in figure 4.5 two optical flow field results obtained with different β parameters. We choose as smoothness parameter $\alpha = 0.01$ along with a relative high parameter $\beta = 0.01$. This is shown in the first image of figure 4.5. The second image shows the flow estimates for the smoothness parameter $\alpha = 0.01$ and smaller control parameter $\beta = 0.00001$. A smaller control parameter leads to larger control values in the minimisation process. As a consequence, smaller control parameters lead to velocity fields which are forced stronger to satisfy the constant-velocity assumption of the Burgers equation. This leads to shock formations of the velocity field at motion boundaries in motion direction and these motion boundaries seem to be well preserved as can be observed in figure 4.5 in the right image.

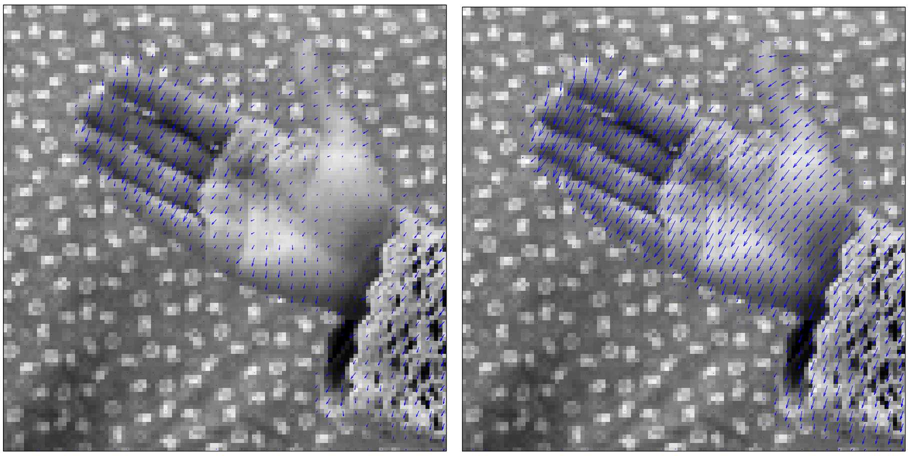


Figure 4.5: Parameter selection. **Left:** Optical flow field with parameters $\alpha = 0.01$ and $\beta = 0.01$. **Right:** Flow field estimate for the parameters $\alpha = 0.01$ and $\beta = 0.00001$. A smaller control-parameter leads to larger control values and therefore, the quality (e.g. sharper motion boundaries) of the flow field improves, due to more control.

However, we observe that the smoothness parameter $\alpha = 0.01$ results in an already homogeneous velocity field but tends to smooth slightly over the motion boundaries.

4.4.3 Experiment with a real image sequences: Control - Force

In this section we illustrate the control behaviour of the optical flow approach for a real-world 2D image sequence with an unexpected motion. The image sequence consists of 10 image frames (200x200) and shows a moving hand which starts to move and then stops again. Figure 4.6 depicts the starting (left column) and stopping (right column) event of the sequence. In the first row the velocity estimates and in the second row the appropriate control/force field is shown. The force field f illustrates the derivation of the expected motion from the observed motion. For the shown example the force field reacts into the opposite direction of the flow field and forces the flow field into the measured state of no motion.

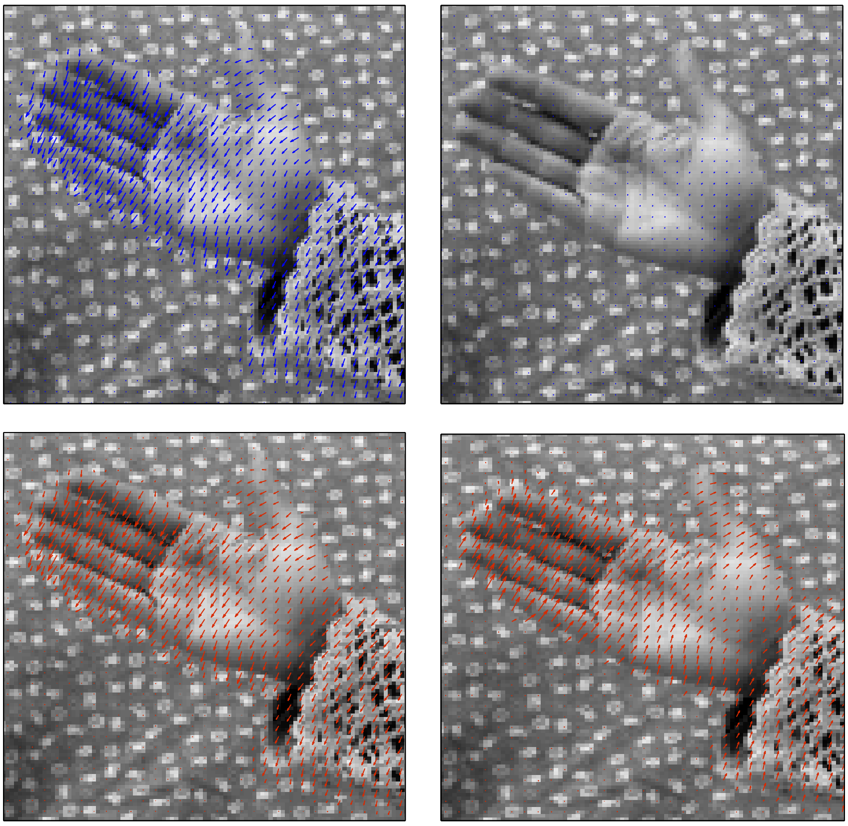


Figure 4.6: "Waving hand" sequence: Unexpected events. **Top:** A waving hand stops. The estimated optical flow field u for a starting (left) and stopping (right) event is depicted in blue. **Bottom:** The corresponding control field f is shown in red. The force acts when the hand starts to move (left) and reacts into the opposite direction of the flow field (right) when it stops and forces the flow field into the observed state of no motion (parameters: $\alpha = 0.01$, $\beta = 0.0001$).

Contrary to the previous approach, which employed a control based interpretation due to the regularisation matching term, this method incorporates a body force directly which corrects the optical flow field in a way, that the flow field is in accordance with the observed image measurements. Here not the deviation between expected and observed image data is penalised rather the force field acts as control quantity in this approach.

4.4.4 Gaussian Noise and Temporal Regularisation

In order to illustrate how our approach copes with noise we used the "waving hand" sequence and added Gaussian noise with standard derivation $\sigma = 20$, as depicted in figure 4.7. The left image shows a single frame of the sequence without noise and the right image with added noise.

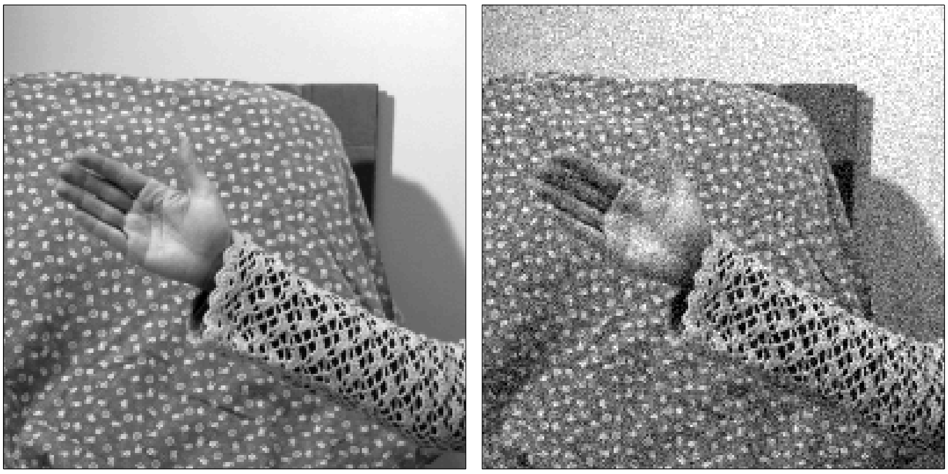


Figure 4.7: The "waving hand" sequence with noise. We added Gaussian noise to the sequence with standard derivation $\sigma = 20$. **Left:** Original grey value image. **Right:** Gray value image with added Gaussian noise $\sigma = 20$.

The optical flow estimate we obtain with our control approach is shown in figure 4.8 (top right). The colour code representation of this velocity field (bottom right) indicates a reasonable estimate for that noise level. However, we observe that the motion boundaries of the velocity field are smoothed out which is a consequence of the higher values for the smoothness parameter that are required to handle the perturbations and to obtain the visually "best" results for this noise level. This is conform to the results we obtained from the noise experiments performed on the "Yosemite" sequence below.

The high noise level leads to the fact that parts of the thumb within the single frames are even for a human undetectable and the computed flow estimate in this region is notably difficult. This is shown in figure (fig. 4.8 left). However, a (more) quantitative evaluation of the control approach for the "Yosemite" sequence with artificially added noise is provided below.

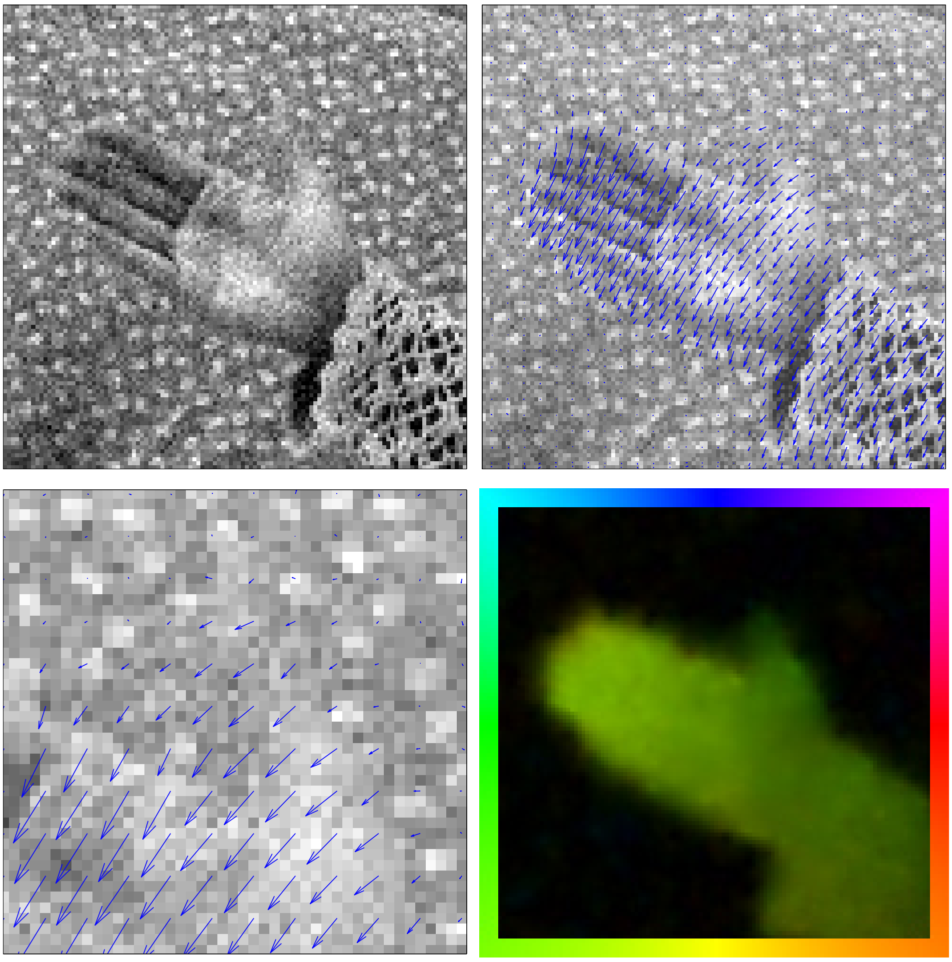


Figure 4.8: Gaussian noise and temporal regularisation. We added noise to the ”waving hand” sequence. **Top left:** Gray value image with Gaussian noise $\sigma = 20$. **Top right:** Optical flow field obtained by our approach (4.3)-(4.4), (parameters: $\alpha = 0.03$ and $\beta = 0.000001$). **Bottom left:** Clipping of the thumb region. The image measurement is highly disturbed and even for a human the original shape of the thumb is undetectable and our obtained flow estimate in this region is inaccurate but due to the temporal regularisation a movement in still computed. **Bottom right:** Colour code representation of our estimate.

”Yosemite” Sequence

In this experiment we investigate the impact of temporal regularisation in our control approach. We chose the ”Yosemite” sequence again as in section 3.4.5 with different Gaussian noise levels $\sigma = 0$, 10, 20 and 40. Based on the results for this we analyse the impact of temporal regularisation. As explained in the previous chapter the computation of (3.4) is done in a sliding window manner and a motion prediction is made for the next frame in the image sequence using the previous results.

Compared to this the control based optical flow approach (4.3)-(4.4) incorporates knowledge about the entire considered image sequence by the means of a forward-backward computation. A visual comparison of the results obtained by the approach (3.4) and the approach (4.3)-(4.4) with a Gaussian noise of $\sigma = 40$ is shown in figure 4.10.

Figure 4.9: Temporal regularisation. We added Gaussian noise with standard derivation $\sigma = 40$ to the "Yosemite" sequence. The shown optical flow field is obtained by the control based optical flow approach (4.3 - 4.4) (parameters: $\alpha = 0.05$ and $\beta = 0.000003$).

The control approach results in a slightly improved vector field. A reason for this can be found in the forward-backward computation, which incorporates additional knowledge of the future frames leading to an improved temporal regularisation. However, we computed the average angular errors for both approaches with increasing noise level. The results are shown in table 4.1.

noise	approach	α	β	RMS	AAE
$\sigma = 0$	Control	0.007	0.0005	0.169	2.88°
	DynMot	0.006	0.00002	0.1784	3.09°
$\sigma = 10$	Control	0.009	0.0001	0.243	4.92°
	DynMot	0.01	0.0003	0.2750	5.68°
$\sigma = 20$	Control	0.02	0.00001	0.3502	6.67°
	DynMot	0.025	0.001	0.3951	7.54°
$\sigma = 40$	Control	0.05	0.000003	0.4968	9.16°
	DynMot	0.05	0.005	0.5226	9.89°

Table 4.1: Performance of our approach (DynMot) in comparison with the Control approach (Control) in presence of noise: We added random Gaussian noise with zero mean and standard deviation $\sigma = 0$, 10, 20, and 40 to the Yosemite image sequence (with clouds). The parameters α and β are optimised for both approaches.

In contrast to the approach discussed in the previous chapter, a higher noise level results in the selection of smaller *beta* regularisation parameters for the control part of the objective functional. The consistently lower error indicates the ability of an improved global motion prediction in our control approach (4.3)-(4.4) exerting a slightly better temporal regularisation. Our explanation for this observation is that the control approach incorporates also future knowledge of the image sequence instead of using only the past information with a prediction in approach (3.4).

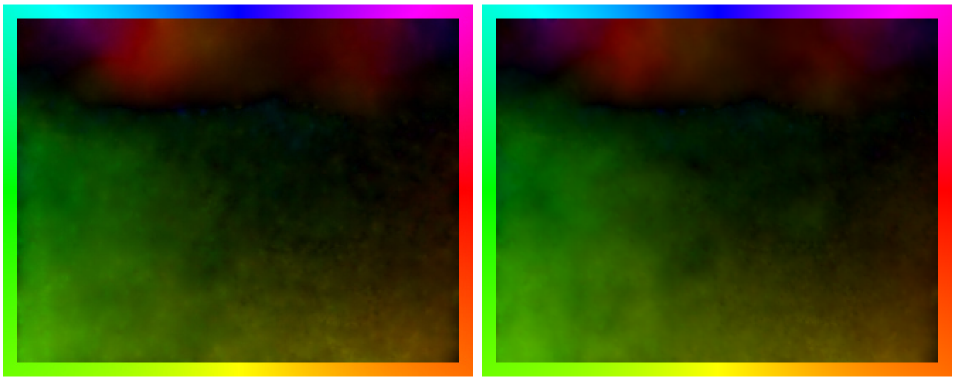


Figure 4.10: Temporal regularisation. Optical flow results for the "Yosemite" sequence with Gaussian noise with standard derivation $\sigma = 40$. **Left:** Colour code representation for the optical flow result obtained by the dynamic optical flow approach (3.4) (parameters: $\alpha = 0.05$ and $\beta = 0.005$). **Right:** Optical flow obtained by the control based optical flow approach (4.3 - 4.4) (parameters: $\alpha = 0.05$ and $\beta = 0.000003$). The error measure (table 4.1) reveals a similar performance slightly in favour of the control approach.

4.4.5 Experiments with real image sequences: Comparison

In this section we intend to compare the results of the spatio-temporal approach of Weickert and Schnörr (batch processing) (2.16) with those of our control approach (4.3)-(4.4). For this we used an image sequence showing a continuous movement with no unexpected motion events. Figure 4.11 depicts a single frame of the sequence showing a slowly downwards moving hand along with the estimated optical flow field. The upper row shows the estimated flow fields using Weickert and Schnörr's batch processing approach and the lower row shows the results of the control approach. We observe, that the control approach computes in regions - that can be considered as more difficult - a smoother flow estimation compared to the spatio-temporal method.

A more detailed zoom in is shown in figure 4.12 and there the hand reveals a "difficult region", which is mainly the gap between the thumb and the hand meaning the gap between the motion boundary of the hand and the motion boundary of the thumb. The spatio-temporal regularisation approach has some difficulties to estimate the optical flow between the thumb and fore finger. In contrast the control approach results in a more reasonable flow estimate in this region. A reason for this behaviour might again lie in the fact that motion boundaries in motion direction of the thumb are better modeled by the Burgers term in combination with the distributed control.

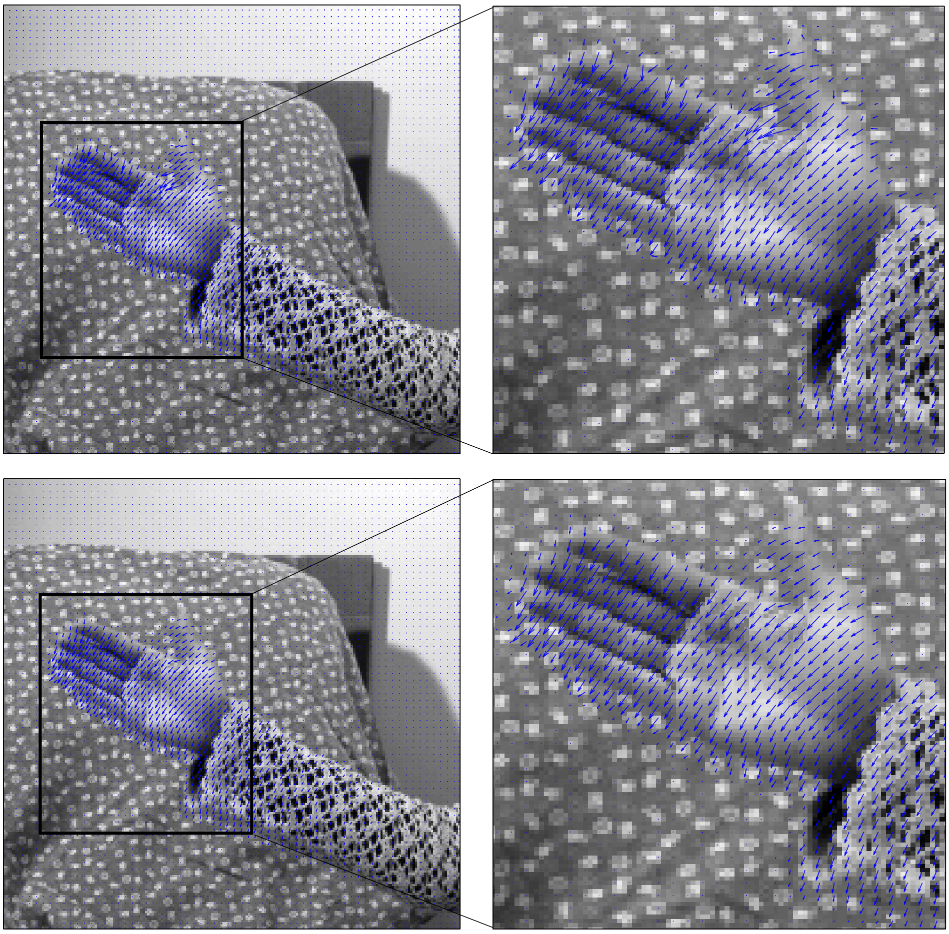


Figure 4.11: Real-world 2D image sequence: "Waving hand" sequence (frame 4 and 5). **Top:** Flow estimation result for the spatio-temporal regularisation approach. **Bottom:** Computational results of our control approach. The control approach computes (in difficult regions) a more reasonable flow estimation compared to the Weickert and Schnörr's method. This is probably due to the better modeling of motion boundaries in motion direction of the thumb and the introduced control.

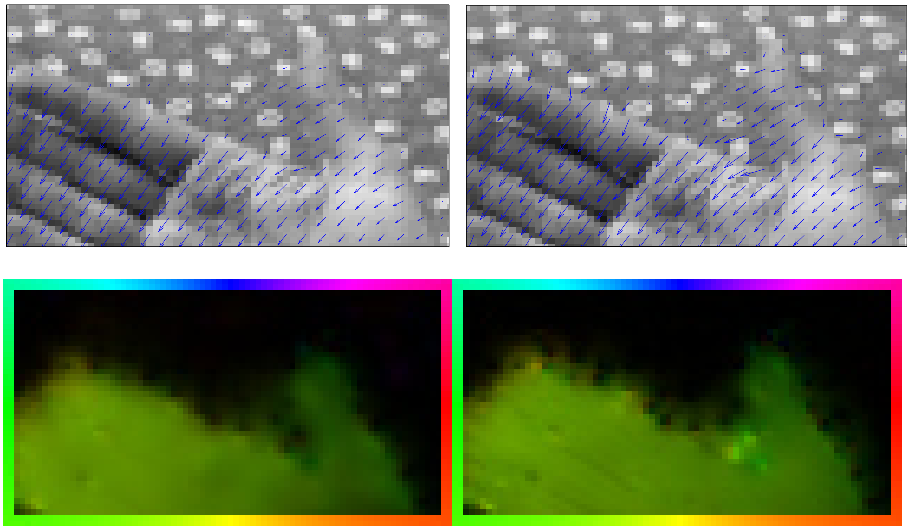


Figure 4.12: Detail of the flow field shown in figure 4.11. **Top left:** Flow field obtained by our approach (4.3)-(4.4). **Top right:** Flow field obtained by the spatio-temporal approach (2.16). **Bottom left:** Colour code presentation for the flow field obtained by approach (4.3)-(4.4) (parameters: $\alpha = 0.01$, $\beta = 0.0001$). **Bottom right:** Colour presentation for the flow field obtained by approach (2.16) (parameter: $\alpha = 0.01$).

Control - Burgers Equation

Our optimal control approach introduces a force (4.4) that acts as a control and has to satisfy the Burgers equation. This force field (depicted in fig. 4.14 and a cutout in fig. 4.13) corrects the optical flow estimate such that it is also in accordance with the observed image data.

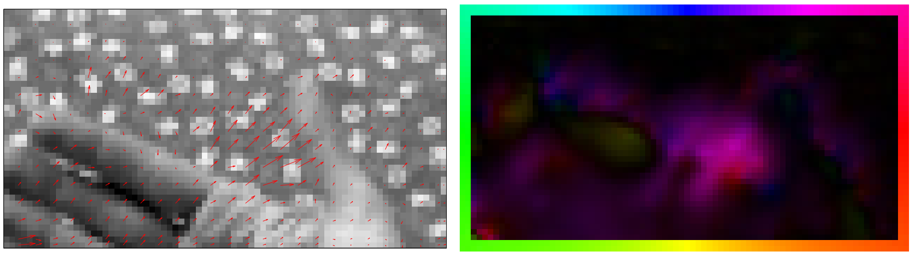


Figure 4.13: Detailed plot of the force field that controls the typical flow estimation near the region which shows the thumb of the hand (parameters: $\alpha = 0.01$, $\beta = 0.0001$).

As depicted in the colour code representation of the control field in figure 4.14 we observe that also at the motion boundary in moving direction the force is acting to correct the flow field and leads to a better motion boundary.

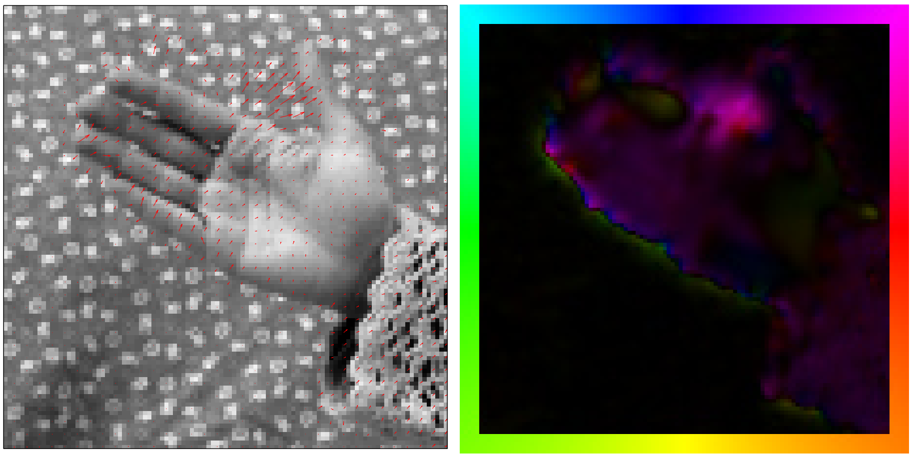


Figure 4.14: Control field (force) obtained by the minimisation of (4.3)-(4.4) (parameters: $\alpha = 0.01$, $\beta = 0.0001$).

4.5 Conclusion

Inspired by our optical flow approach of the previous chapter we explored here the idea to trim the approach to an optimal control approach to dynamic image motion. The resulting approach represents a control formulation for variational motion estimation. We minimise - similar to the previous approach - the objective functional subject to a physical constraint. However, instead of having a matching term within the objective function the control approach introduces a force which pushes the vector field directly towards a vector field that is in accordance with the image data and also satisfies the Burgers equation with the smallest amount of force.

We solve the variational optical flow approach within a control framework using *physical* prior knowledge with a forward-backward iterative computation scheme. The optical flow field should satisfy the Burgers equation, which describes the acceleration of moving particles. In order to estimate the specific flow field of apparent velocities of moving objects in an image sequence, control variables are included and determined by minimising an appropriate objective function which relates the flow field and the control variables to the given image sequence data.

In our experiments we exemplarily showed that our implementation presents a reasonable convergence characteristics and an acceptable number (about 500 for this shown example) of iterations are sufficient for the computation of the flow. However, our straight forward implementation leaves room for improvements. The main features we observed within our control approach are already present in the approach proposed in the previous section. The motion boundary is modeled due to a shock forming sharply in motion direction but smooth in the rarefaction area. The introduced control can be used as indicator for unexpected motion events. Compared with the previous approach we observed an improved capability of the motion estimation for image sequences that are disturbed with a high noise level. However the overall performance is - as far as our experiment allow to conclude - similar.

An interesting question for further work is if the reverse of the image sequence could be exploited to improve the computation of the motion boundary in the rarefaction area as the reversal should lead to shocks in this part then. Another idea for further work is to investigate whether the approach can be stated in a receding horizon control formulation.

Part II

Image and Video Inpainting

5

Image Inpainting

In the previous chapters we were concerned with variational approaches for motion estimation using physical prior knowledge. We found that the Burgers equation represents a beneficial regularisation term for a dynamic description of motion due to its modelling of transport processes. In this part of the thesis we investigate if this prior has advantages for other applications of computer vision as well. In particular for inpainting applications, where motion estimation and transportation processes of image information can be used to recover missing parts in images or image sequences. Therefore, we first present some well-known inpainting approaches for still images which provides us also with some basics used in our video inpainting approach proposed in chapter 6 where we make use of a fluid dynamic based information transport process.

5.1 Introduction

The idea of inpainting has its origin in methods that were used for the restoration of artwork from the Renaissance. The restoration of damaged paintings was done manually by professional artists [44, 139]. The main idea is to use the information around the damaged area of an image to restore the damaged part. That often means that the image colour in the part that has to be repaired should be a smooth continuation of the boundary colour with preserving incoming edges. The mechanism of ”inpainting” is also used by photographers and film restaurateurs. Inpainting techniques regained interest particular in the field of computer vision and the media industry as it turned out to be useful for digital images and image sequences too and has led to several applications. For example it plays a role in film industry to obtain ”special effects” by removing auxiliary devices from the scenes. Other applications includes scratch removal, the reconstruction of occluded image parts, or the removal of undesired objects.

A famous inpainting approach was introduced to image processing by Bertalmío, Sapiro, Caselles and Ballester [10], where they proposed a fundamental mathematical formulation of the inpainting problem with a third-order PDE. Since then several different approaches have been developed: This includes fluid dynamic based approaches [9, 57], variational [27], Bayesian [96], and curvature based approaches [26, 135].

In computer vision the aim of inpainting is to find a reasonable and undetectable reconstruction of image parts where the image information is damaged or even completely missing. An example is shown in figure 5.1 where the black bars should be reconstructed by using the information that is still available to obtain an image that should be as close as possible to the original image.

Figure 5.1: Inpainting problem: The task is to recover the missing information (black bars) in the image by using the image information that is still available.

5.1.1 Problem

In our following discussions we use the following notation: Let $I_0 = I_0(x)$ be an ideal image which is defined in the whole image region $x = (x_1, x_2) \in \Omega_0$.

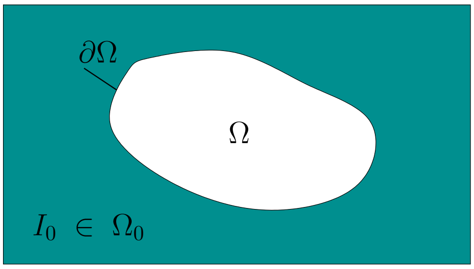


Figure 5.2: Topology of the inpainting model. The inpaint region $\Omega \subset \Omega_0$ is to be reconstructed using the image information $I_0 \in \Omega_0$ that is still present outside of the boundary $\partial\Omega$ of the inpaint region.

Damaged or missing regions which may result from scratches, acquisition errors, occlusion or other mechanisms define the so-called inpaint region $\Omega \subset \Omega_0$ (cf. fig. 5.2) in which the image information should be restored. We assume that such subregions are without any image information itself. The image intensity I_0 is assumed to be smooth outside of Ω and it is assumed that the image data is known on the boundary $\partial\Omega$. The goal is to recover the image in the inpaint region using the surrounding information in an undetectable manner.

5.1.2 Classification

In the literature one can find several different inpainting methods. A classification of these methods could be made based on the application or on the kind of inpainting techniques that are used. In this thesis we favour a classification which is based on the technique underlying the individual approaches (fig. 5.3).

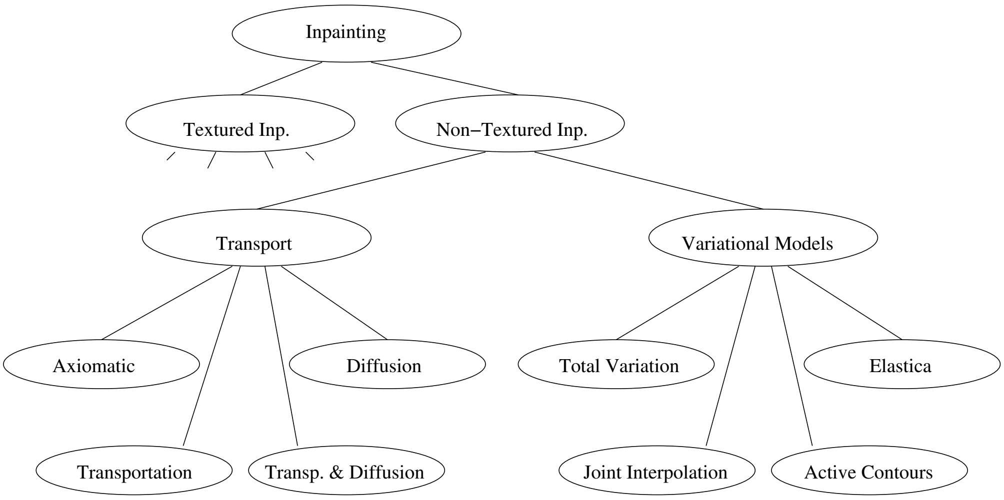


Figure 5.3: Inpainting classification based on the technique underlying the individual approaches.

We distinguish between texture based and non-texture based inpainting approaches. In this thesis we do not consider texture based inpainting methods which synthesise stochastically the reproduction of a texture from a sample within the inpaint region. Usually, a statistic model is proposed first, followed by pixel and patch-based sampling techniques for the actual texture inpainting. For texture based inpainting approaches which are, however, beyond the scope of this thesis, the interested reader may be referred to [71, 140, 11, 39, 55].

The non-texture based inpainting approaches we are interested in are based on PDEs and are intended mainly for images of man-made objects as natural images inevitably involves their statistical modelling. We distinguish further between transport mechanisms and variational based methods.

The first category includes the approaches which explicitly transport information into the inpaint region using the surrounding data. This transport processes can be modelled - for example - by transport equations that occur in fluid mechanics: convection or diffusion processes [10, 29] or their combination [9, 28].

The second category include approaches that made use of variational models, approaches which utilise the total variation measure [30, 27, 113, 114], elastica models [26, 90], joint interpolation methods [5] and Mumford-Shah image models [98, 46, 24].

In the following we present some established inpainting approaches mainly with the aim to provide ideas for our video based inpainting approach in chapter 6.

5.2 Established Techniques

The inpainting approaches we describe in the following come from both the transport based and the variational based categories. Our focus lies on the modelling of physical based fluid transport mechanisms and variational formulations in anticipation of the development and description of a new video inpainting approach in chapter 6.

We start in the next section with a combined transport and diffusion process based image inpainting approach, which describes fundamental inpainting techniques used in our video inpainting approach. The following approaches utilise physical equations and the total variation measure known from approaches to image denoising and decomposition [113, 114, 4, 25, 138].

5.2.1 Fluid Mechanic Based Approach

In the following we are concerned with the inpainting approach of the authors Bertalmío, Bertozzi and Sapiro [9]. They make use of ideas from classical fluid dynamics in order to propagate image information along isophote lines by a transport equation from the exterior into the inpaint region Ω.

An important point of this fluid dynamic based image inpainting approach is that it links the image intensity function I with the velocity field $v(x,t) = (v_1(x,t), v_2(x,t))^\top$, $x = (x_1, x_2)$ which is supposed to be divergence-free and therefore satisfies the condition $\nabla \cdot v = 0$.

Such a vector field can be obtained from the scalar image intensity function $I = I(x,t)$ by using the following equations

$$\begin{aligned} v_1 &= -\partial_{x_2} I, \\ v_2 &= \partial_{x_1} I \end{aligned}$$

or shorter $v = \nabla^\perp I$ where v_1 and v_2 are the two components of the vector field $v = (v_1, v_2)^\top$. The iso-intensity lines of the image intensity function I are also known as isophote-lines of the image - in fluid dynamics they are called streamlines. In this approach the image intensity function plays the role of a stream function whose isophote-lines define the stream lines of the flow. In figure 5.4 these lines together with the appropriate vector field v are illustrated

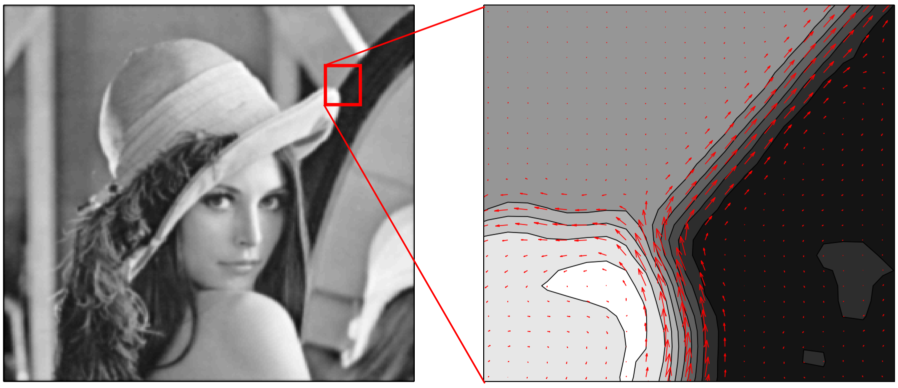


Figure 5.4: Isophote lines **Left:** Smoothed "Lena" image. **Right:** Isophote lines within a part of the image overlaid with the transport velocity field $v = \nabla^{\perp} I$ which is parallel to the isophote lines.

for an image. The vector field along the isophote lines is then used to transport image information into the inpaint region. But instead of transporting the grey values directly the Laplacian ΔI of the image is transported. The vorticity transport equation which can be derived from the Navier-Stokes equation is used by the authors for the propagation process as there is a connection between the vorticity s and the Laplacian of an image. This can be seen if we recall that the vorticity for an incompressible flow is defined as the rotation of the corresponding velocity field. For the two-dimensional flow we obtain

$$s := \nabla \times v = \partial_{x_1} v_2 - \partial_{x_2} v_1 = \partial_{x_1}^2 I + \partial_{x_2}^2 I = \Delta I, \tag{5.1}$$

which means that the Laplacian of I itself represents the vorticity. The transport process is illustrated in figure 5.5. The Laplacian of the image ΔI is

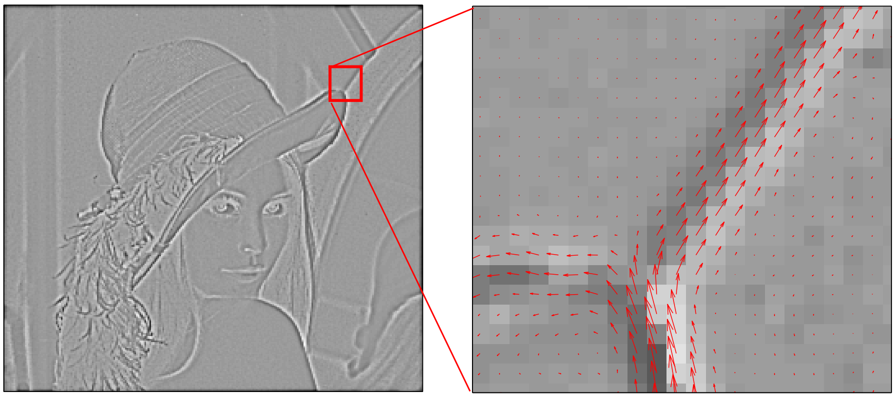


Figure 5.5: Vorticity. **Left:** Laplacian of the Lena image ΔI as vorticity s, which is transported with the velocity field v in isophote direction. **Right:** A section of the left Laplacian image overlaid with the transport velocity field $v = \nabla^{\perp} I$ in isophote direction.

transported using the velocity field $v = \nabla^{\perp} I$ along the level-lines into the inpaint region. Note that this process transports only the information of the Laplacian into the inpaint region and the actual intensity has to be recovered by integrating the appropriate Poisson equation with respect to the intensity information of the boundary of the inpaint region.

After this short description of the approach we provide details of our implementation below.

Approach

The principle of the inpainting approach discussed here is based on fluid dynamical transport mechanisms. This transport mechanism satisfies a particular partial differential equation which describes incompressible flows that are used to transport image information into the inpaint region. Incompressible flows are governed by the Navier-Stokes equations. Note that a flow is incompressible if the velocity field u is divergence-free $\nabla \cdot u = 0$ and the mass is conserved. The Navier-Stokes equation for incompressible flows reads as follows

$$\partial_t u + (u \cdot \nabla)\, u = -\frac{1}{\rho} \nabla p + \nu \Delta u \quad \text{in } \Omega, \tag{5.2}$$

along with the constraint for incompressible fluids

$$\nabla \cdot u = 0, \tag{5.3}$$

and the boundary condition

$$u = 0 \quad \text{on } \partial\Omega.$$

In these equations ν denotes the coefficient of kinematic viscosity, ρ represents the density and p the scalar valued pressure of the fluid.

In the following the aim is to apply the Navier-Stokes equation to digital images which consist of scalar pixel values. In order to do so we make use of a reformulation of the incompressible Navier-Stokes equation (5.2) that is explained in section 7.4 in detail. The resulting equation depends only on the velocity u and no longer on the pressure p. This special case of the incompressible Navier-Stokes equation is known as the *vorticity transport equation* which represents also a form of an advection-diffusion equation (cf. also sec. 7.3 and 7.2):

$$\begin{aligned} \partial_t \omega + (u \cdot \nabla)\omega &= \nu \Delta \omega && \text{in } \Omega, \\ \omega &= 0 && \text{on } \partial\Omega. \end{aligned} \tag{5.4}$$

Here the vorticity ω in two dimensions is defined as the following scalar valued measure

$$\omega = \nabla \times u = \partial_{x_1} u_2 - \partial_{x_2} u_1 = \partial^2_{x_1} \phi + \partial^2_{x_2} \phi = \Delta \phi$$

and provides the connection between the incompressible velocity field u and a scalar stream function ϕ, which represents the image intensity in this approach. In the following we list the variables used in the inpainting approach and their analogous counterpart in fluid dynamics:

Inpainting Quantities	Fluid Dynamic Quantities
image intensity I	stream function ϕ
isophote direction $v = \nabla^{\perp} I$	fluid velocity u
smoothness $s = \Delta I$	vorticity $\omega = \Delta\phi$

The authors of [9] introduced the following modified version of the vorticity transport equation:

$$\begin{aligned} \partial_t s + (v \cdot \nabla)s &= \nu \nabla(g(|\nabla s|)\nabla s) \quad \text{in } \Omega, \\ s &= \Delta I_0 \qquad\qquad\qquad \text{on } \partial\Omega. \end{aligned} \tag{5.5}$$

this means we use $s = \Delta I$ outside of the boundary to construct a continuity solution on the boundary, to transport the information continuously into the inpaint region. The function g allows for anisotropic diffusion [141] within the smoothness term. For $g = 1$ equation (5.5) is equivalent to equation (5.4). See also section 7.2.1.

Using the modified vorticity transport equation (5.5) the image information is – as desired – transported into the inpaint region. The vorticity transport equation is solved for a two-dimensional velocity field $v = (v_1, v_2)$ with a given vorticity $s = \Delta I$ on the boundary $\partial\Omega$ of the inpainting domain Ω. This transports the vorticity information into the inpaint region Ω using the surrounding data.

Finally, the image intensity I, which defines the transport velocity field $v = (-\partial_{x_2} I, \partial_{x_1} I)^{\top}$, is recovered by solving the Poisson equation

$$\begin{aligned} \Delta I &= s \quad \text{in } \Omega, \\ I &= I_0 \quad \text{on } \partial\Omega, \end{aligned} \tag{5.6}$$

with fixed Dirichlet boundary conditions (using the original image data I_0 at the boundary). We refer to section 7.1 for a detailed description of the Poisson equation.

To summarise, the vorticity transport equation (5.4) is derived from the Navier-Stokes equation and an analogous form of the equation is given by (5.5), where the isotropic diffusion term is modified to allow anisotropic filtering. By using equations (5.5) the vorticity s (5.1) is transported into the inpaint region which is then used to update the image intensity by integrating the Poisson equation (5.6).

Algorithm

In the following, we illustrate our implementation of the fluid mechanical based inpainting algorithm described above. We first apply it to a simple example problem which is shown in figure 5.6 in order to show its principle behaviour. There, the aim is to reconstruct the missing information in the (black) inpaint region. The user interaction that is required within the algorithm is to mark the inpaint regions.

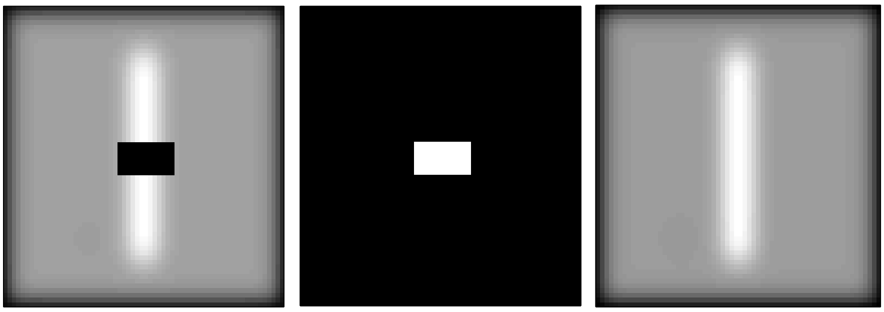


Figure 5.6: Left: Image with missing information (black rectangle in the middle). **Middle:** The inpaint region is indicated by a binary mask. In this case the value 1 (white) represents the inpaint region. **Right:** Inpaint result.

The vorticity transport equation (5.5) is solved within the inpaint region Ω using appropriate boundary values. The inpaint algorithm is sketched in (Algorithm 2) and further explained below. For details of the numerical implementation of the partial differential equations we refer to the appropriate sections in part III of this thesis.

Algorithm 2 Navier-Stokes Inpainting Algorithm

```
Require: choose tolerance $\epsilon$, viscosity $\nu$,
compute initial vorticity $s_0$ (5.7)
compute initial velocity $v_0$ (5.8)
repeat
    solve (5.5) for vorticity $s$ using 8.2.3 in $\Omega$
    solve (5.6) using 8.1.5 to obtain updated $I_m$ in $\Omega$ using (Algorithm 3)
    compute vorticity $s$ from $I_m$ in $\Omega$
    compute velocity $v$ from $I_m$ in $\Omega$
until $|I_m - I_{m-1}|/|I_m| < \epsilon$
```

Initialisation: The inpaint region Ω in the input image $I_0 = I_{t=0}$ has to be marked by an appropriate binary inpaint mask as indicated in figure 5.6. In our implementation the inpaint region Ω is represented by ones and the other regions by zeros. A tolerance value ϵ has to be defined which defines when the solution is considered to have reached a steady state solution. A value for the viscosity coefficient ν must be chosen, which weights the strength of the smoothness term. Higher values of ν result in a higher diffusion rate (as in sec. 7.2). For our test values $\nu = 0.02$ and $\epsilon = 10^{-6}$ turned out to be reasonable choices.

To generate the initial vorticity s_0 the Laplacian ΔI_0 of the image I_0 is used. It is computed over the full image region Ω_0 Ω with continuously boundary conditions on $\partial\Omega$

$$\begin{aligned} s_0 &= \Delta I_0 && \text{in } \Omega_0 \\ s_0 &= \Delta I_0 && \text{on } \partial\Omega. \end{aligned} \tag{5.7}$$

The initial velocity field $v_0 := v$, $v = (v_1, v_2)^\top$ is obtained using

$$\begin{aligned} v &= (-\partial_{x_2} I_0, \partial_{x_1} I_0)^\top && \text{in } \Omega_0 \\ v &= v_0 && \text{on } \partial\Omega. \end{aligned} \tag{5.8}$$

The numerical realisation of the Poisson equation is explained in section 8.1.5. An example for such an initial vorticity s_0 along with the velocity field v_0 is depicted in figure 5.7.

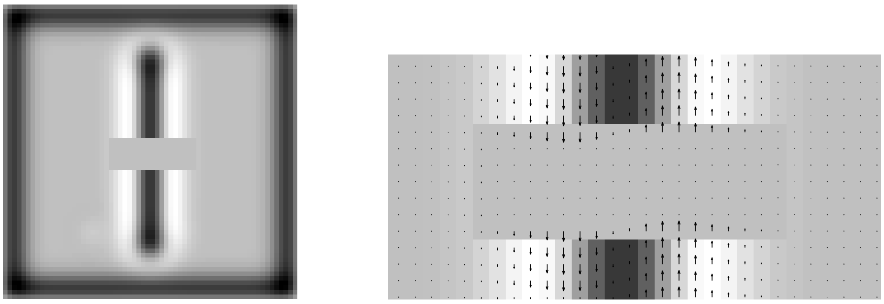


Figure 5.7: Left: Laplacian image ΔI_0 of the image in figure 5.6 as initial vorticity. **Right:** Part of the Laplacian image along with its initial velocity field.

This initialisation is used to enter the *main loop* of the algorithm:

The vorticity transport equation (5.5) is solved within the inpaint region Ω (line 4 of algorithm 2). Numerical details of our implementation can be found in 8.2.3.

The transported vorticity information within the inpaint region is used to obtain the updated image intensity I_m by solving the Poisson equation (5.6) with Dirichlet boundary conditions (line 5) for which we used the preconditioned conjugate gradient method explained in section 9.1.

From image I_m, velocity values v and vorticity values s for the next iteration $m + 1$ are computed. Figure 5.8 depicts the computed vorticity and the corresponding velocity after 50 iterations.

The algorithm stops when the relative change of the image intensity is below the threshold ϵ indicating that the image I changes not appreciably (cf. line 8).

For our small example a steady state solution is obtained after 150 iterations which is shown in figure 5.6.

Note, that we perform 10 iterations without and 5 iterations with anisotropic diffusion on the intensity image I_m leading to sharper edges. (See also sec. 7.2.1 for anisotropic diffusion.)

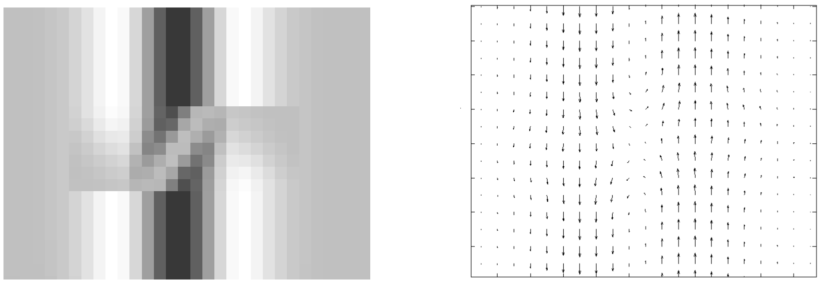


Figure 5.8: Left: The result of the vorticity transport after 50 iterations. **Right:** Velocity field after 50 iterations.

In figure 5.9 and 5.10 the results for a more interesting inpaint example is shown. They illustrate the inpaint process for a small region which contains edges and a junction within the "Lena" image. In figure 5.9 the transport of the Laplacian is indicated while figure 5.10 shows the recovered inpaint region. (Another result is shown in figure 5.15 later in this chapter.)

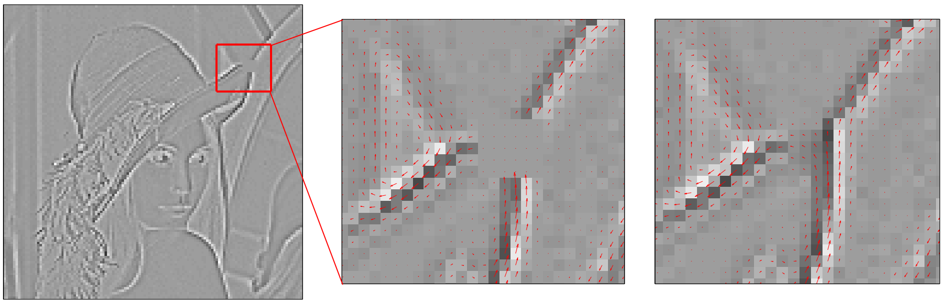


Figure 5.9: Left: Laplacian of the "Lena" image 5.10. **Middle:** Cutout of the inpaint region along with the initial velocity field v_0. **Right:** Computational result of the vorticity transport after 50 iterations.

5.2.2 Total Variation Image Inpainting

In this section we present a variational inpainting approach introduced by Chan and Shen [27] that is known as total variation (TV) image inpainting. This approach can be seen as a generalisation of the total variation model for denoising of Rudin, Osher and Fatemi [114] as for an empty inpainting region they turn out to be the same.

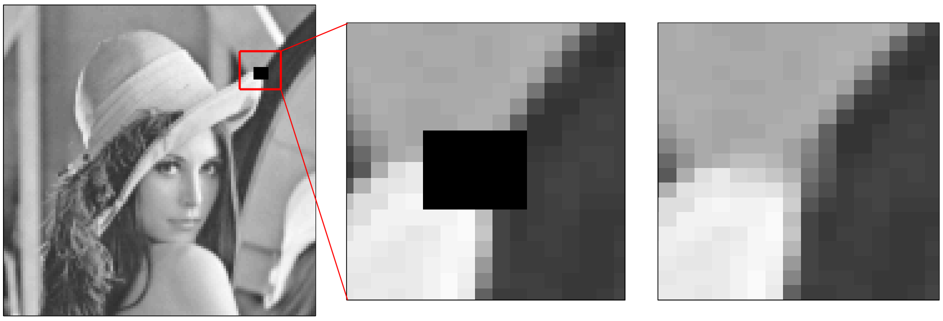


Figure 5.10: Left: "Lena" image. **Middle:** Cutout of the inpaint region. **Right:** Recovered image region.

The following topology which is sketched in figure 5.11 is used for the inpainting model. The area Ω_B surrounds the inpaint region Ω. The boundary $\partial\Omega$ of the inpaint region lies in the interior of $\Omega \cup \Omega_B$, which we also call extended inpaint region.

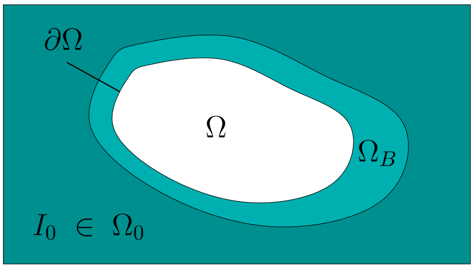


Figure 5.11: Topology used within most variational inpainting models. The inpaint region Ω is to be reconstructed using the image information I_0 that is present in the band Ω_B around the inpaint region Ω.

The TV-inpainting problem is formulated as the minimisation of the following energy functional with respect to I:

$$J(I) = \int\limits_{\Omega \cup \Omega_B} |\nabla I| \, dx + \frac{1}{2}\lambda_B \int\limits_{\Omega_B} |I - I_0|^2 \, dx, \tag{5.9}$$

where

$$\lambda_B = \begin{cases} \lambda, & x \in \Omega_B \\ 0, & x \in \Omega \end{cases} \tag{5.10}$$

plays the role of a Lagrange multiplier which incorporates the denoising on the surrounding image region on Ω_B. Thus the aim of the approach is to find an image I on the extended inpaint region that minimises (5.9) thereby using the total variation (TV) image prior

$$TV(I) = \int\limits_{\Omega \cup \Omega_B} |\nabla I| dx \tag{5.11}$$

within the energy functional (5.9). The TV-norm is essentially a L_1-norm of the image derivative and therefore a measurement of the amount of occurring oscillations of the image function I. Compared to a prior using the L_2-norm, this prior tends to penalise discontinuities in I less severely and therefore helps to preserve the edges of the original image.

The Euler-Lagrange equation for the energy functional (5.9) results in

$$-\nabla \cdot \left(\frac{\nabla I}{|\nabla I|_a} \right) + \lambda_B (I - I_0) = 0, \tag{5.12}$$

for the extended inpaint region along with von Neumann boundary conditions on $\partial(\Omega \cup \Omega_B)$. Here for numerical reasons the authors used $|\nabla I|_a := \sqrt{|\nabla I|^2 + a^2}$ to replace $|\nabla I|$ introducing the "lifting" parameter a. According to the authors the parameter leads to an improved conditioning of the model and represents a threshold determining where the model tries to imitate the harmonic inpainting $|\nabla I| << a$ and where the model resumes the TV inpainting $|\nabla I| >> a$.

A steepest-decent algorithm for the objective functional (5.9) could be implemented stepping iteratively in the direction:

$$\partial_t I = \nabla \cdot \left(\frac{\nabla I}{|\nabla I|} \right) + \lambda_B (I_0 - I). \tag{5.13}$$

However, in [27] the authors describe in detail the preferable implementation resulting in an iterative algorithm solving the Euler-Lagrange equation (5.12) directly.

An advantage of this inpainting model is that it can deal with noisy images but the same authors formulated the curvature-driven diffusion inpainting model [29] to overcome some drawbacks they observed in the TV-inpainting model. Their main point is that it violates the connectivity principle [1] of visual perception (cf. fig. 5.12).

5.2.3 Curvature-Driven Diffusion Inpainting

The Curvature-Driven Diffusion (CDD) inpainting model [29] is based – as the name implies – on a diffusion mechanism. It is a modification of the Euler-Lagrange equation used in the TV-inpainting model of Chan and Shen [27] and incorporates an anisotropic diffusion mechanism which is driven by the curvature information of incoming isophote lines.

As already mentioned, the TV-inpainting model described in the previous section cannot successfully connect isophote lines that are far separated. In particular the connectivity principle cannot be fulfilled when the length of the inpaint

---

[1] The Connectivity Principle refers to the fact that our visual system is able to connect parts of objects which are separated from each other by some occlusion or foreground regions [77, 105].

region Ω is larger than the width of the object of interest as indicated in figure 5.12. The CDD model is able to overcome this drawback.

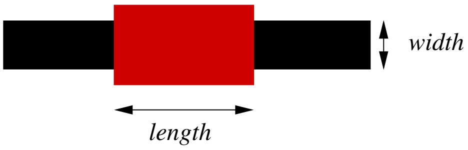


Figure 5.12: Connectivity principle of visual perception: The CDD inpainting model can connect an object that is occluded (by the inpaint region) even if the width of the object is smaller than the width of the inpaint region (red area).

This model is driven by the following equation:

$$\partial_t I = \nabla \cdot \left(\frac{g(|\kappa|)}{|\nabla I|} \nabla I \right) + \lambda_B (I - I_0), \tag{5.14}$$

with λ_B as extended Lagrange multiplier as defined in (5.10). For the initialisation, the information of the original image is used in Ω_B and arbitrary values can be utilised in the inpaint region Ω (an initialisation with the TV-inpainting results are recommended in [27]):

$$I(x, 0) = I_0(x), \quad x \in \Omega_B \cup \Omega.$$

In the TV model, the diffusion coefficient D_{TV} depends reciprocally on the gradient of the image I:

$$D_{TV} = \frac{1}{|\nabla I|}. \tag{5.15}$$

This means that the diffusion in smooth regions is much higher than in regions with edge structures. Chan and Shen modified the inpainting model in such a way that the diffusion coefficient D_{CDD} depends on the curvature of the isophotes of the image I:

$$D_{CDD} = \frac{g(|\kappa|)}{|\nabla I|}. \tag{5.16}$$

The curvature κ along any isophote line of an image I is given by

$$\kappa = \nabla \cdot \left(\frac{\nabla I}{|\nabla I|} \right), \tag{5.17}$$

An example of this curvature measure is illustrated in figure 5.13. For $|\nabla I| = 1$ this results in the curvature $\kappa = \Delta I$ which is just the Laplacian of the image. The function g is assumed to be continuous. In order to suppress large curvatures and support small curvatures the authors identified the following requirements for g:

Figure 5.13: Illustration of the curvature measure (5.17) for the "Lena" image.

$$g(s) = \begin{cases} 0, & s = 0 \\ \infty, & s = \infty \\ between, & 0 < s < \infty. \end{cases}$$

The introduction of $g(\kappa)$ penalise large curvatures and encourage small ones. That means, that the diffusion gets stronger where the isophotes have larger curvatures, while the diffusion decreases in regions with straight lines. Replacing g with 1 results in the classical TV inpainting model.

The CDD inpainting model is a third-order PDE model and is able to continue isophote lines and allows to connect long distances by straight lines satisfying the connectivity principle. Figure 5.14 presents the computational result for the variational model approach (5.14). The fourth-order elastic model described in the next section is designed to allow the continuation of isophote lines with a curvature.

Comparison

Figure 5.15 presents the result for the fluid dynamic based approach (5.5)-(5.6) and the variational model approach (5.14). For this example it seems that the Navier-Stokes inpainting results in a visually better inpainting.

For our purposes we implemented the CDD and the Navier-Stokes inpainting approaches and we will compare in chapter 6 some results of our video inpainting with the frame-by-frame extension of the Navier-Stokes inpainting approach [9]. However, in the following we sketch the ideas of some other inpainting approaches.

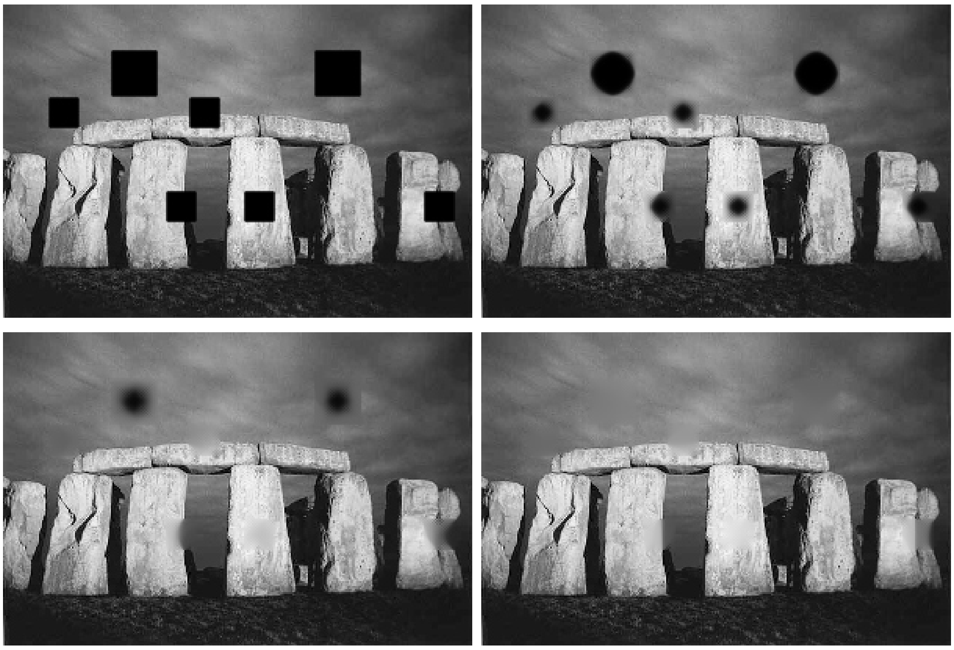


Figure 5.14: Inpainting example "Stonehenge": **Top left:** Image with inpaint regions with zero as initialisation (black regions). **Top right:** Inpaint result for iteration 90 with the CDD approach (5.14). **Bottom left:** Inpaint result for iteration 240. **Bottom right:** Final result after convergence. Smooth regions like the sky are well recovered, while regions with edges and junctions (edges of the stones) tend to be smoothed out.

Figure 5.15: Left: "Lena" image with marked inpaint regions (black bars). **Middle:** Inpainting result using the Navier-Stokes inpainting approach (5.5)-(5.6). **Right:** Inpainting result using the Curvature Driven inpainting approach (5.14). The Navier-Stokes inpainting results in a visually better inpainting.

5.2.4 Elastica based Inpainting Model

In 1744 Euler studied a particular class of smooth curves and their application to shape modelling under external forces [106]. These kind of curves are nowadays known as Euler elastics. Birkhoff and de Boor used them in a non-linear spline model [13]. The first author who introduced this mathematical knowledge as a prior curve model to the computer vision community was Mumford [97]. Masnou and Morel [90] enhanced this model in their elastica based variational inpainting approach to interpolate isophotes using dynamic programming. In the following we are concerned with an approach of Chan, Kang and Shen [26] who modified the model of Masnou and Morel in a variational and PDE based context resulting in the elastica and curvature inpainting approach. It is designed to overcome the drawback of Chan and Shen's model which approximates the isophotes only as straight lines. The elastica model interpolates the two boundary end points of an isophote line by an Euler elastica, therefore as a curve.

Recall that the model of Bertalmío et al. transports information along isophotes, while the CDD (and TV) inpainting model of Chan and Shen diffuses information across the inpainting domain.

Chan, Kang and Shen [26] applied the elastica image model

$$\int_{\Omega} (a + b\kappa^2)\, dx$$

in order to improve their TV inpainting model resulting in the Euler elastica inpainting approach where the following energy functional

$$J(I) = \int_{\Omega \cup \Omega_B} (a + b\kappa^2)\, |\nabla I|\, dx + \frac{1}{2}\lambda \int_{\Omega_B} |I - I_0|^2\, dx \tag{5.18}$$

has to be minimised. Here a and b are two positive weights.

The elsastica based model results in the TV-inpainting model of Chan and Shen [27] for the extreme case when $\frac{a}{b} = \infty$. The Euler-Lagrange equation of the energy functional (5.18) is denoted by the evolution equation

$$\partial_t I = \nabla \cdot \left((a + b\kappa^2) n - \frac{2b}{|\nabla I|} \partial_\tau (\kappa |\nabla I|)\, \tau \right) - \lambda_B (I - I_0), \tag{5.19}$$

where $n = \frac{\nabla I}{|\nabla I|}$ the normal vector and $\tau = n^\perp$ the tangential vector.

In [26] the authors describe the implementation of the evolution equation (5.19) resulting in a central and upwind differencing scheme with min-mod discretisation.

The elastica inpainting approach combines the transportation mechanism of Bertalmío's model [9] and the curvature-driven diffusion mechanism of Chan and Shen's model [29].

5.2.5 Ginzburg-Landau Equation

Grossauer and Scherzer presented in [57] an inpainting approach that is based on the complex Ginzburg-Landau equation. This equation was originally developed by Ginzburg and Landau [53] in order to describe physical phase transition phenomena near their critical temperature. The equation also turned out to be useful to describe chemical reactions, pattern and shock developments, and boundary layer problems. Grossauer and Scherzer described an analogy between physics and image analysis where they interpret the homogeneous areas as domains with nearly constant grey value intensity and the phase transition regions as edges in the image.

The equation is able to produce homogeneous regions which are separated by phase transition regions. These transition regions are usually represented by minimal surfaces, that is surfaces with vanishing mean curvature.

The authors minimise the following energy functional

$$J(u) = \frac{1}{2} \int\limits_{\Omega} \left\{ |-i\nabla u|^2 + \alpha |u|^2 + \frac{1}{2}\beta |u|^4 \right\} \, dx \tag{5.20}$$

and note that the $-i$ in the first term is for their approach a non-essential left over from quantum mechanics. Here $u \in \mathbb{C}$ is a complex valued function on the image domain Ω. The authors link the real valued part of u with the grey value intensity I of an image. With the choice $-\alpha = \beta = \frac{1}{\epsilon^2}$ they discretised the differential equation

$$\partial_t u = \Delta u + \frac{1}{\epsilon^2} \left(1 - |u|^2\right) u \quad \text{in } \Omega \tag{5.21}$$

in order to find a minimum of the functional (5.20) using Dirichlet boundary conditions . The equation (5.21) is known as an instance of a reaction-diffusion equation and the parameter ϵ is interpreted in physics [42] as the coherence length and corresponds to the width of the transition region (for images this is the width of the edges).

In [57] the numerical solution is obtained by an explicit forward finite difference scheme. According to the authors this inpainting approach provides very good results for locally small inpaint regions and is suitable for the removal of cracks or text in images. The authors provided also an example for the restoration of high-dimensional data using a generalised form of the Ginzburg-Landau equation which is useful for frame interpolation and the reconstruction of fragmentary given surfaces.

5.2.6 TV-Stokes Equation

The inpainting approach of Tai, Osher and Holm [129] is related to the transport based approaches of Bertalmío et al. [9, 5] and the minimisation of the divergence of an vector field like in [48]. In the following we describe the approach of Tai, Osher and Holm in more detail. Their inpainting method is divided into two steps. The first step consists in the computation of a vector field within the inpaint region that is related to the isophote directions.

In order to propagate the normal vectors $n = \nabla I_0$ of the isophote lines or equivalently the tangential vectors $v = \nabla^\perp I_0$ of the isophote lines into the inpaint region the authors propose to minimise the following energy functional:

$$J(v) = \int\limits_{\Omega} |\nabla v| dx + \frac{1}{2}\lambda \int\limits_{\Omega_B} |v - v_0|^2 dx \tag{5.22}$$

subject to

$$\nabla \cdot v = 0,$$

as they need a divergent free vector field for the second step. They discretised the corresponding Euler-Lagrange equation

$$\begin{aligned} -\nabla \cdot \left(\frac{\nabla v}{|\nabla v|}\right) + \lambda_B (v - v_0) - \nabla \mu &= 0 \quad \text{in } \Omega \cup \Omega_B, \\ \nabla \cdot v &= 0 \quad \text{in } \Omega \cup \Omega_B, \\ \nabla v \cdot n_{out} &= 0 \quad \text{on } \partial(\Omega \cup \Omega_B) \end{aligned}$$

and used them in a gradient descent algorithm to obtain the optimal v. Here n_{out} denotes the outward unit normal vector of the extended inpaint region $\Omega \cup \Omega_B$. In [129] they sketched their implementation of this step for a simplified situation assuming that Ω_B is just one pixel wide.

The second step of their algorithm consists of the reconstruction of the image I from the vector field in the inpaint region by minimising the following energy functional

$$J(I) = \int\limits_{\Omega} \left\{ |\nabla I| - \nabla I \cdot \frac{n}{|n|} \right\} dx + \frac{1}{2}\lambda \int\limits_{\Omega_B} |I - I_0|^2 dx.$$

where the vectors n are perpendicular to the vectors v obtained in the previous step, i.e. $n = v^\perp = (-v_2, v_1)^\top$. The Euler-Lagrange equations are

$$\begin{aligned} -\nabla \cdot \left(\frac{\nabla I}{|\nabla I|} - \frac{n}{|n|}\right) + \lambda (I - I_0) &= 0 \quad \text{in } \Omega, \\ \left(\frac{\nabla v}{|\nabla v|} - \frac{n}{|n|}\right) \cdot n_{out} &= 0 \quad \text{in } \Omega_B. \end{aligned}$$

Again, the implementation of this step for a simplified situation can be found in [129]. The results presented in that of this paper indicate that the approach can deal with rather large inpaint regions. Using different boundary conditions for the inpaint region Ω the authors were able to influence the information which is propagated into the inpaint region.

However, this approach is related to our video inpainting approach in the next chapter by using a two step algorithm. In the first step a transportation field is recovered by an variational approach. In the second step the image information is reconstructed within the inpaint region. While the authors minimise an appropriate energy functional with respect to the image function, we transport the image information by using a physically transport equation into the inpainting region.

5.3 Discussion

In this chapter we presented some established inpainting approaches which make use of fluid dynamically inspired transport mechanisms and approaches that use the variational framework in order to model the inpaint process. All the approaches have in common that they are implemented by solving the corresponding partial differential equations incorporating some prior knowledge about the desired outcome of the inpainting. We were mainly interested in the approaches which use physical mechanisms like (anisotropic) diffusion and the vorticity transport equation.

Inspired by these approaches we develop in the next chapter a new inpainting approach for image sequences using the Burgers equation and an advection-diffusion equation. There these transport and diffusion mechanisms are used as prior knowledge to complete missing parts in video sequence.

The main difference between video inpainting and inpainting of still images is that for video inpainting the information of the entire image sequence can be used. However, a simple way to apply the image inpainting approaches presented in this section to video inpainting is to use them frame-by-frame, which means that each frame of the sequence is treated as an independent image, dropping any information from the other frames.

6

Video Inpainting

In part I of this thesis we presented variational based optical flow algorithms for motion estimation in image sequences. In chapter 5 we were concerned with image inpainting algorithms using fluid dynamical processes to reconstruct missing parts in images. In this chapter we want to combine this knowledge in order to design a new algorithm for the reconstruction of missing data in image sequences. Our focus is on exploring the potential and the limitations of this PDE-constrained approach to inpainting rather than on working out application software based on existing related work.

6.1 Introduction and Motivation

Video inpainting has the aim to recover video information in damaged (or with logos or text occluded) parts of image sequences. Figure 6.1 depicts an example

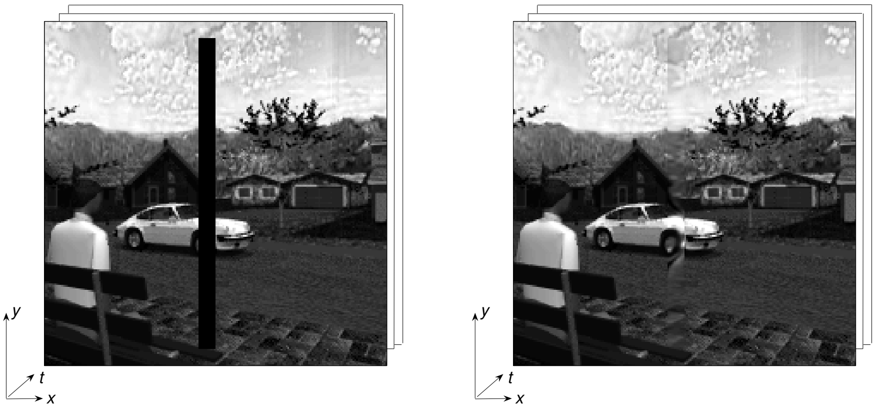


Figure 6.1: Video inpainting scenario. **Left:** Lost image information is represented by the black bar in the shown image frame. We intend to recover the missing parts in the image sequence using the surrounding image information of the image frames. **Right:** Computational result obtained by our video inpainting approach described in this section. This result will be discussed in detail below.

where an area of video information is artificially removed. The regions of missing image data should be reconstructed in a visually undetectable manner.

Image inpainting as described in the previous chapter is used as a foundation for video inpainting. One of the first video inpainting approaches was introduced in [9] as a frame-by-frame based method, where the image inpainting method described in section 5.2.1 is used to recover the missing information frame-by-frame. In this way every image inpainting algorithm can be used for video inpainting. Partial differential equations in these methods are applied only spatially and do not take into account the temporal information that a video provides. It is therefore often restricted to applications which only needs to recover image information for small missing image parts.

Therefore, it is desired to design video inpainting approaches, which also exploit the knowledge that the data is temporally continuous. There are several options to incorporate temporal information into inpainting approaches. For example, temporal filtering applied only to each pixel along the temporal axis of image sequences [31] or spatio-temporal filtering in combination with motion estimation techniques [81, 83], where the filtering effects pixels and neighbour pixels, which belong to the same object in every image. The video inpainting approach presented in this chapter takes into account spatio-temporal filtering based on physical model equations. We will utilise PDE based techniques that restores degraded image sequences with large spatial and temporal missing data areas. Our video inpainting approach consists of two steps. In the first step we recover the motion vector field for a subsequent pair of image frames using the dynamical flow estimation approach described in chapter 3. The obtained flow field is then used in the second step to transport the image information into the missing image parts using an advection-diffusion equation.

The novelty of this approach is the combination of two PDE-based transport mechanisms to reconstruct the video information in a spatio-temporally manner leading - to a certain degree - to a structure preserving inpainting. The approach exploits the constant velocity assumption of the Burgers equation and the assumption that a grey value of an object doesn't change along its motion trajectory.

Within our experimental section we show the advantages of our video inpainting approach along with its limitations.

6.1.1 Overview

In subsection 6.1.2 we first present some work related to our approach. We introduce our two step video inpainting approach in section 6.2 where we explain the interaction of the two employed partial differential equations. In section 6.3 we provide details of our implementation. In section 6.4 we illustrate our results for several synthetic image sequences and for real image data. We used an extension of the Navier-Stokes image inpainting approach 5.2.1 to video inpainting to compare the results with our new approach. Within the conclusion in section 6.5 we provide some innovative ideas for further work.

6.1.2 Related and Further Work

Like our approach one of the first video inpainting approaches presented by Bertalmío et al. [10, 9] belongs to the class of PDE based approaches. They extended their Navier-Stokes image inpainting, which we presented in section 5.2.1 to a frame-by-frame video inpainting method and applied the partial differential equations only in a spatially context. That means the PDEs are solved separately for each frame to complete the image sequence frame-by-frame. Contrary to this approach, we incorporate spatio-temporal information of the video sequence. Ideas from classical fluid dynamics are also used to transport exterior image information into the region of missing data. The authors use also a two steps process. Note that our spatial temporal regularised approach recovers in the first step the velocity estimate between subsequent image frames using the optical flow approach presented in chapter 3. Then in a second step the estimated motion field is used to transport the image intensities surrounding the marked region of missing data into the inpaint region. In Bertalmíos et al. inpainting approach, the image information is first transported and then the image measurements are recovered. However, this process is only done frame-by-frame and is therefore a spatial regularised approach.

Grossauer [56], who is also concerned with an optical flow and PDE based video inpainting approach used the estimated optical flow fields to detect degraded regions in the image sequence and rebuild the detected regions. Grossauer makes use of a variational formulation for the optical flow problem as well, but instead to minimise the quadratic form of the optical flow constrain equation (2.2) he minimises the displacement field between pairs of consecutive image frames. Spatial smoothness is achieved by a homogeneous regularisation. Corrupted regions/pixels are determined by a threshold parameter. The difference between the image intensity in the actual frame and the corrected image intensities of the previous image frame, using the backward and the forward flow field, are computed. While Grossauer uses the displacement vector fields to copy the image information into the detected regions, we use a physically model equation to transport the available image information into the inpaint regions.

Another optical flow based approach was presented by Kokaram [82] using spatio-temporal information which allows to restore small missing parts in noisy image sequences.

Another related problem where it is useful to exploit spatio-temporal continuation is the task of automatically filling in missing spatio-temporal parts in video sequences which might be caused by the removal of undesired objects. In [144] Wexler modifies the model [10, 9] to a global optimisation problem to obtain global consistency. The resulting is designed to satisfy the local spatio-temporal constraint that similar areas in the video sequence should be similar to related parts of the image sequence. The global component of the algorithm results in the fact that all the local patches are consistent in a global space-time manner. Wexler applied the approach to texture synthesis in three dimensions as well, but this is according to the author computationally quite intensive. One of the latest work is from Patwardhan [109], which extends an earlier optical flow based video inpainting approach [108] that is based on the idea to separate foreground and background objects.

In the following we introduce our inpainting approach based on a global optimisation problem to recover the optical flow of the sequence. This is then used to transport the image information to reconstruct the sequence.

6.2 Video Inpainting Approach

We present in this chapter a video inpainting approach exploiting two physical motivated transport equations. Our two-step approach recovers the optical flow field of the image sequence in the first step and transports the image information along with the flow field into the inpaint region in the second step. The flow field is computed using the dynamic image motion approach of chapter 3. The image information is transported using an advection-diffusion equation as described below.

Similar to the previous chapter we assume that the regions of missing information (inpaint region) in the video are known and indicated by a binary mask. As we focus on the reconstruction of missing parts in the video we further simplify the problem by using the same mask for each frame in the image sequence, that means that the region of missing information is always at the same location for each image frame of the sequence. To obtain the inpaint region automatically, Grossauer [56] as mentioned before used the estimated optical flow fields itself to detect the degraded regions in the image sequences. However, we just assume that the inpaint region is known and given by the appropriate mask.

In the following Ω represents the inpaint region of the image frame and Ω_0 represents the entire image domain, including the inpaint region with respect to time t. A particular pixel grey value is defined by its location and time and denoted as $I_0(x_1, x_2, t)$. As the image frames are usually recorded at discrete times within the time interval $[0, T]$ the time is sometimes given by its frame number $1, ..., N$.

In the following we describe the two successive steps of our inpaint algorithm.

6.2.1 First Step: Optical Flow Computation

The first step of our algorithm has the aim to compute the motion field between a subsequent pair of image frames of the image sequence. The resulting vector field is then used in the second step as the transportation field which transports the video information into the inpaint regions. The motion estimation is obtained by minimising the following energy functional

$$\begin{aligned} J(u) \;=\; & \frac{1}{2}\int\limits_{\Omega_0} \Big\{ \Big(\partial_t I_0 + u \cdot \nabla I_0 \Big)^2 \\ & + \alpha \Big(\nabla u_1^\top D(\nabla I_0) \nabla u_1 + \nabla u_2^\top D(\nabla I_0) \nabla u_2 \Big) \\ & + \beta \| u - u_T \|^2 \Big\} dx, \end{aligned} \tag{6.1}$$

which is essentially the same as in (3.4) with the matching term - including the solution ($u_T := u(x,T)$) of the Burgers equation

$$\begin{aligned} \partial_t u + (u \cdot \nabla) u &= 0, \\ u(x,t)_{t=0} &= u_0, \end{aligned} \tag{6.2}$$

as regularisation term. We minimise the energy functional (6.1) in the time interval $[t_n, t_{n+1}]$ which represents here the time interval between a subsequent pair of image frames. Note, the difference between the energy functional (3.4) and (6.1) consists in the definition of the image intensity function I respectively I_0. Here the function I_0 refers to the image measurements including the inpaint region. We solve the objective function over the entire image domain including the region of missing image measurements. As already shown in section 3.4.3 the dynamic optical flow approach recovers due to the smoothness constraint velocity estimates in regions of missing image measurements as well. As a result we obtain a velocity field which is further used as a transport field to transport the image information into the inpaint region.

The author of [56] uses a variational formulation for the optical flow problem, but instead to minimise the quadratic form of the optical flow constraint equation (2.2) he minimises the displacement field between two consecutive image frames. Spatial smoothness is achieved by a homogeneous regularisation. Grossauer uses the displacement vector fields just to copy the image information into the inpaint regions, while we use a physically based transport mechanism to transport image information from the surrounding area into inpaint regions as described below.

6.2.2 Second Step: Image Information Transport

Recall, Bertalmío et al. [10, 9] interpreted the image intensity as a stream function of an incompressible flow, the Laplacian of the image intensity as vorticity of the fluid and the vector field of transportation by the stream function itself. The vorticity transport equation, a special case of the Navier-Stokes equation is used in a modified way to transport the Laplacian of the image intensity (which equates to the vorticity) into the inpaint region. Then in a following step the Poisson equation is solved to recover the image intensity of the image.

In contrast to the transport of the Laplacian along the level lines, we transport the image measurements itself from the exterior into the region of missing image data. For this purpose we employ the standard advection-diffusion equation along with the optical flow estimates of the image sequence. The optical flow field u which is obtained by the minimisation of (6.1) is used to transport the appropriate image information of the surrounding areas into the inpaint region. For this task we have to solve the following advection-diffusion equation

$$\begin{aligned} \partial_t I_0 + u \cdot \nabla I_0 &= \nu \Delta I_0, \\ u(x,0) &= u_0 \quad \text{in } \Omega,\ t = 1, ..., N, \\ I_0 &= g \quad \text{on } \partial\Omega,\ t = 1, ..., N \end{aligned} \tag{6.3}$$

within the inpaint region. The coefficient ν represents the kinematic viscosity of the fluid function which has some influence on the smoothness of the image intensities in the inpaint region. Structures which are transported into the inpaint region tend to appear smoother with higher values of ν. Note that we are using Dirichlet boundary conditions to solve (6.3). In the following we describe our numerical implementation of the approach.

6.2.3 Algorithm

In this section we describe how we implemented the full video inpainting approach. However, for particular details of the implementation we refer to the appropriate sections in the numerical part of this thesis. First we are concerned with the initialisation therefore gathering the appropriate start data for the computation in the first frame:

a) Initialisation

  i) Mark the inpaint region Ω within the given video sequence I_k, $k = 0, ...N$.
  ii) Solve the energy functional (6.1) to obtain the minimiser $\hat{u}_0$ between the first two image frames I_0 and I_1 in Ω_0. Note the $\hat{u}_0$ provides a vector field also within the inpaint region.
  iii) Within equation (6.3) the minimiser $\hat{u}_0$ is used to transport the image information into the inpaint region Ω. This can be performed with (6.3). Therefore the inpainted version I_0^{inp} of the image can be computed from the image I_0. We denote with I^{inp} the images within the inpaint process.
  iv) Set $I_1 = I_0^{inp}$.

In the following we discuss how the next frames are processed. The iterative computations are quite similar to the computations needed for the first frame.

b) Main loop

  i) Set $k = k + 1$.

ii) Solve the energy functional (6.1) to compute the velocity field u_k between I_k and I_{k+1} in Ω_0.

iii) Use the resulting minimiser $\hat{u}_k$ to transport the image information from the surrounding area into the inpaint region Ω. Use the advection-diffusion equation (6.3) to obtain the inpainted image I_k^{inp} from the image I_k.

iv) Set $I_{k+1} = I_k^{inp}$ and repeat the main loop until $k = N$.

6.3 Numerical Solution

As our video inpainting approach is a two step approach we have to solve the flow estimation part (6.1)-(6.2) and the image data transportation part (6.3) successively. For the solution of the optical flow estimation part we refer to section 3.3 and 8.3, of the numerical part of this thesis.
In section 8.4 we describe the numerical implementation of the advection-diffusion equation (6.3) - which is used to transport the image measurements from the surrounding area into the inpaint region - using a finite difference method.

6.4 Experiments

In this section we test and evaluate our algorithm using synthetic and real image sequences. First, we explain the process of information transport into the region of missing image data (inpaint region). We investigate advantages, disadvantages and limitations of our approach. We do this by applying the video inpainting approach to different motion patterns and to a real image sequence. After that, we compare the results with a *frame-by-frame* modification of the Navier-Stokes image inpainting approach from section 5.2.1. In our experiments we simply used a fixed inpaint region which is unchanged for each image frame of the entire sequence.

The parameters for the solution of (6.1) - (6.2) are selected due to the experience from chapter 3. The viscosity term of equation (6.3) is set to $\nu = 0.02$ within all experiments. Higher values lead to an over smoothing of the image structures.

6.4.1 Information Transport

Experiment: ”Street” Sequence

Our first experiment demonstrates how image information is transported into the inpaint region. First, we investigate the synthetic ”street” image sequence [93], known from section 3.4.4. The inpaint region is illustrated by a black bar (with size 180x10 pixels as depicted in the top left image of figure 6.2).

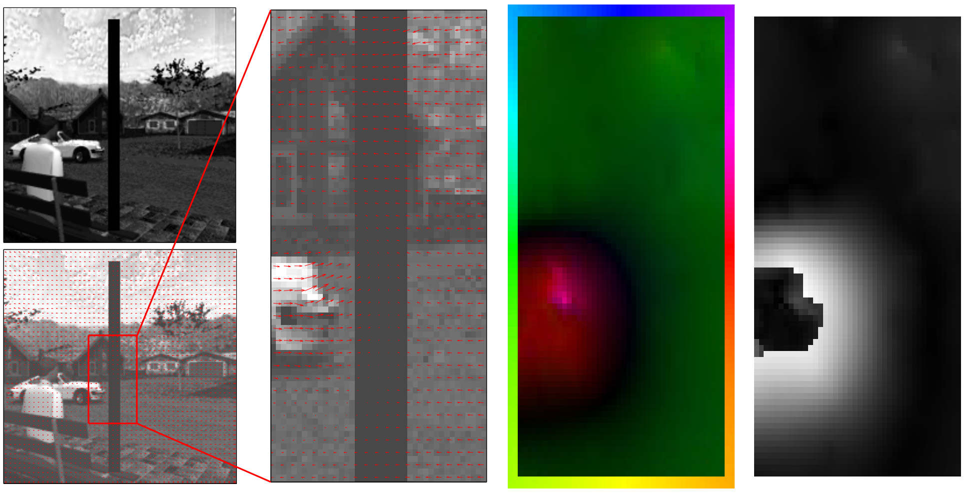


Figure 6.2: Inpaint example using the "street" sequence. **Top left:** One frame within the street sequence along with the inpaint region marked by a black bar. **Bottom left:** Velocity estimates obtained by (6.1) - (6.2) (parameters: $\alpha = 0.01$, $\beta = 0.001$). **Second:** Cutout of the inpaint region along with the velocity estimate. **Third:** Colour representation of the velocity field for the cutout. **Fourth:** RMS error plot of this region with respect to true and estimated velocity field.

In the first step of our inpainting algorithm we solve (6.1) - (6.2) for the first two subsequent image pairs. Due to the spatial smoothness regularisation the estimated flow field extends into the homogeneous inpaint region. In fact we exploit the undesired effect of the blurring over motion boundaries. Neighbouring movements around the inpaint region determine the flow field in this region. As a result we can "reconstruct" a dense flow field between the first two image frames in the region of missing image measurements. The inertia modelled by the Burgers equation results in a velocity field that is transported by itself over time and the surrounding velocity is transported into the inpaint region as illustrated in figure 6.3.

In regions with uniform flow (upper half area of the right image in fig. 6.2) the velocity field is well estimated (cf. RMS error plot comparing the original motion with the estimated motion in right image of fig. 6.2). Using this image flow along with the advection-diffusion equation (6.3), image information is transported into the inpaint region resulting in a desired reconstruction.

From chapter 3 we know, that due to the motion of the camera – which follows the car – the background moves in the opposite direction of the car. This motion pattern represents the main difficulty of this sequence: The two motion areas are moving against each other and result in a shock formation within the solution of the Burgers equation. Therefore, the most interesting region is the motion boundary of the car.
The inpainting results for that region are shown in figure 6.3 for different image frames. Note, that the shock formation can be observed at the front of the car.

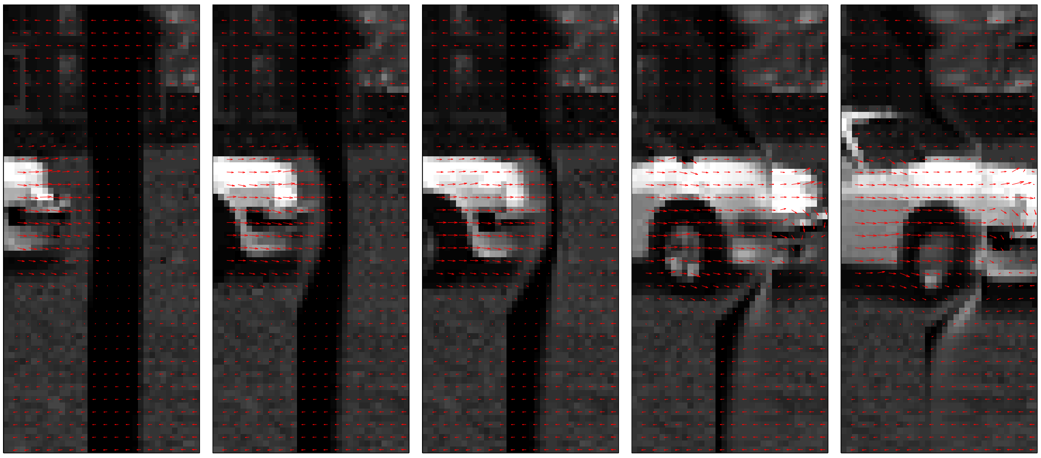


Figure 6.3: Inpaint results at different times (frames). Influenced by the Burgers equation the velocity field is transported by itself within (6.1) - (6.2) and with the velocity field the image information is transported into the inpaint region using the advection-diffusion equation (6.3).

Starting with the first image in figure 6.3 we observe that the dense flow field in the upper part of the image within the inpaint region moves to the left, while in the middle part two motion fronts (one from the left and the other coming from the right) are moving against each other. The velocity field is used to transport it by itself due to the Burgers equation and can be observed in the second image.

In particular, the velocity field - resulting from the minimisation of (6.1) - (6.2) - transports the image data into the inpaint region by using the advection-diffusion equation (6.3). Investigating the fourth image and the last image of the figure shows, that our transportation strategy allows an image structure preservation (wheel of the car) over the transportation process. Image information is transported from the right (upper and lower part of the image) and from the left (middle part of the image) into the inpaint region and leads after a few time steps to the reasonable reconstruction of the video data.

Experiment: "Sphere" Sequence

For the "sphere" sequence we know from section 3.4.4 that the motion field is a static velocity field. Like before the inpaint region is given in each frame (top left of fig. 6.4) and the computed flow field in the inpaint region leads to a dense flow field. The image information is transported by the counter clockwise rotational velocity field into the inpaint region. We depict the progress of image information transportation in figure 6.4. The inpainting results after solving (6.3) using the velocity result obtained by solving (6.1)-(6.2) are shown.
Using the recovered velocity field the image measurements are transported into the inpaint region. The transportation process can be best recognised in the second and third image frame of the sequence (top centre and top right). The velocity near the top of the sphere is lower and therefore the amount of image

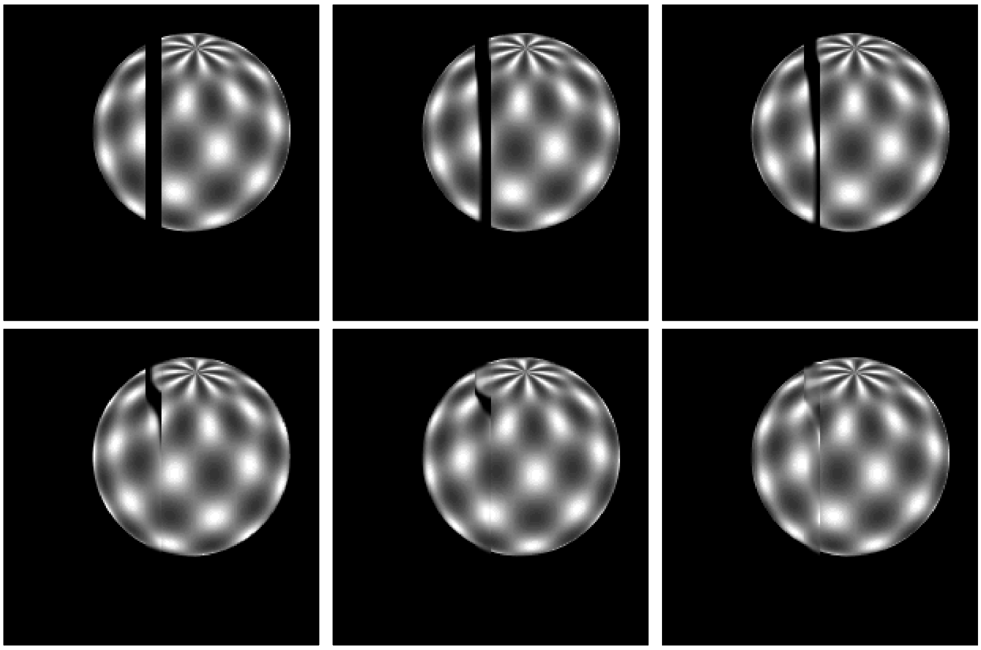


Figure 6.4: Inpainting example: A sphere rotates in front of a stationary background. Image transportation process. The image information is transported with the velocity field obtained by solving (6.1)-(6.2) in time. The image measurements around the inpaint region (marked with a black bar) in the first frame (top left) are transported using this velocity field in time. This means the result of this transportation can be recognised in the second frame (top centre). The velocity near the top of the sphere is lower and therefore the amount of image information that can be transported into the inpaint region is reduced. With an increasing time (and so in frame number) the last gap can closed as shown in the last image.

information that can be transported into the inpaint region is limited. With increasing time (and frame-number) the gap is completely closed as shown in the last image. However, some small artifacts in the structure remain visible.

6.4.2 Breakdown Points: Illustrative Experiment

The following experiment enables us to determine the benefits and drawbacks of our approach. We choose the ”office” sequence [93] as known from section 3.4.4. The inpaint region is represented by a black spiral as depicted in figure 6.5 (left), and is for each frame the same. We would like to point out that we choose an image sequence with divergent motion pattern on purpose as it reveals the limitations of our approach more easily.

The ”office” sequence can be considered to be challenging due to the violation of the grey value consistency equation (2.2) induced by the divergent motion pattern (cf. sec. 3.4.4). Due to the chosen inpaint region, additional boundary areas are included and previous seen image measurements vanish and unseen parts appear. The flow estimate that is used to transport the image information

Figure 6.5: Challenging experiment. **Left:** Image frame with (inpaint) region of missing image data marked by a black spiral. Due to the inpaint region, additional boundary areas are included and image measurements vanish into or appear from the inpaint region. The "office" sequence is challenging due to the violation of the grey value consistency equation and the divergent motion pattern. **Right:** The underlying optical flow field – which results from the first step of the computation algorithm – is a divergent velocity field. The velocity field within the black homogeneous inpaint region is reconstructed due to spatial regularisation (dense flow, cf. also experiment 3.4.3).

into the inpaint region is shown in figure 6.5 (centre and right). The inertia behaviour of the Burgers equation lead to the transport of the velocity by itself into the inpaint region (cf. also experiment 3.4.3).

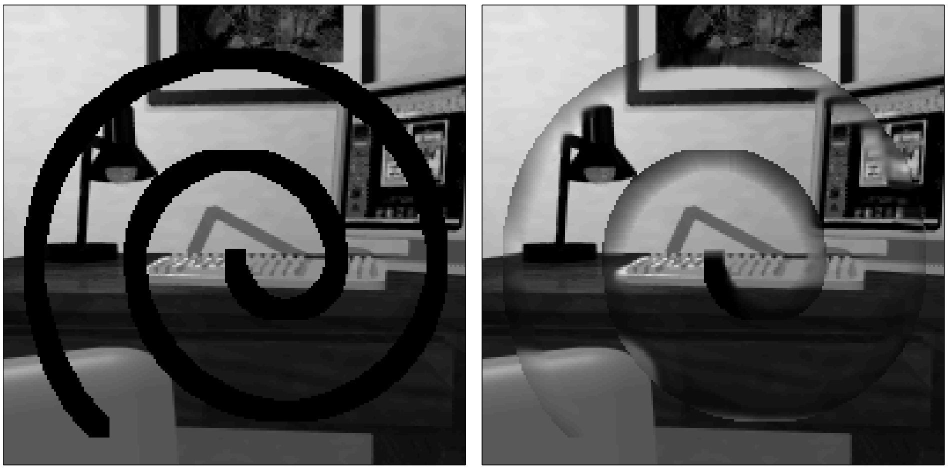


Figure 6.6: Image reconstruction with a divergent transport vector field. **Left:** "Office" frame with inpaint region (black spiral). **Right:** Inpainting result. If the image information is transported with a divergence velocity field, the image intensity decreases within the inpaint region due to the deformation of the image structure.

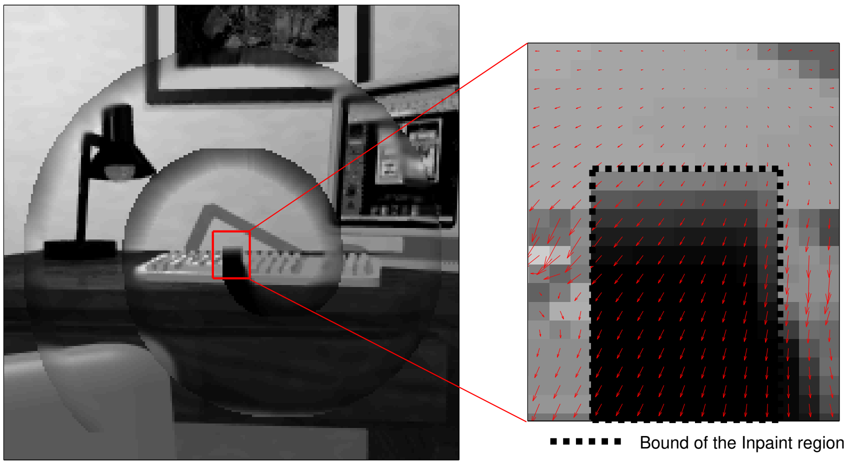


Figure 6.7: Inpaint region at the centre (projection of the optical axis of the camera) of a divergent motion field. **Left:** Inpaint result for this ”office” frame. **Right:** Velocity estimates for the inpaint region. We obtain for this region very small velocity estimates. The transportation of image information into this region is very slow and the reconstruction of the image at this point is therefore challenging.

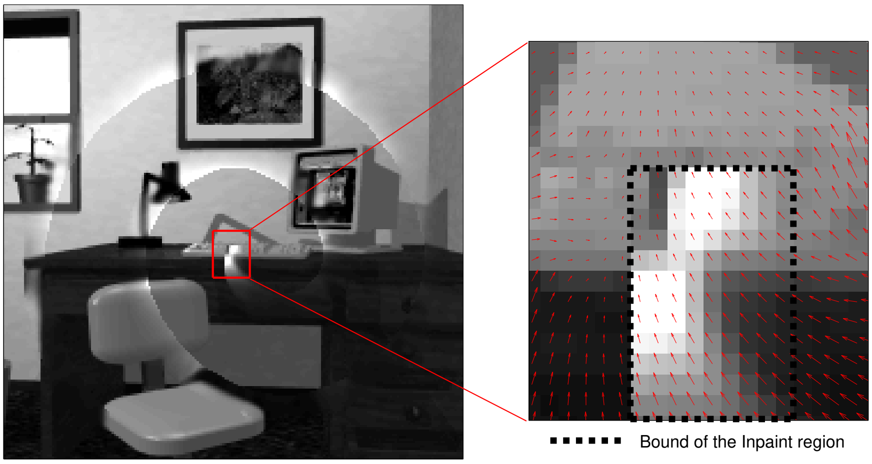


Figure 6.8: Inpainting results for the reversed image sequence. Starting the computation again for this image sequence allows the transportation of image information from the other direction into the inpaint region. **Left:** Inpaint result for the reversed ”office” sequence. **Right:** Cutout from the centre of the inpaint result. The reversed sequence lead to the opposite situation in the centre of the image. The white areas indicate a compression of the grey values. Image information is transported from the surrounding region into the inpaint region and is compressed, resulting in higher image value measurements.

If the image information is transported with an outwards directed divergence velocity field, one can observe that the image intensity decreases towards the outward inpaint region. This is due to the divergent vector field which spreads the image information outwards, as shown in figure 6.6 (right). Therefore, this image parts have a lower image intensity (darker) or are even without any image information as shown in the detailed figure 6.7 of this scene.

As the optical centre (projection of the optical axis of the camera to the image) of this sequence is fixed, we obtain only very small velocity estimates for this region. As a consequence, the transportation of image information is very small near to this point and nearly no inpainting is performed.

The inpaint boundary region in the opposite direction of the flow does not get the reasonable image information from the surrounding area. However, using the reversed image sequence (now a "zooming out" scenario) the transport process is reversed as well and the image information is transported in the opposite direction. Results for this can be seen in figure 6.8.

The reversed version of the sequence leads to the opposite situation in the centre of the image (cf. fig. 6.8) and in other parts of the inpaint region as well. Image information is transported from the outward regions into the inpaint region leading to a compression of grey values resulting in brighter image value measurements.

Nevertheless, this challenging example indicates also that our approach is able to preserve image structures. In figure 6.9 we show some details of the inpaint result that shows the advantage of the transport processes used in the approach. The information from previous image frames is transported into the inpaint area over time preserving the structure of the objects.
We note that it is possible to preserve the structures in inpaint regions, due to the fact that the velocity field is locally relative homogeneous (with nearly the same direction and velocity).

6.4.3 Real Image Sequence Inpainting

In this section we show the results of our approach for a real video sequence recorded in our Mannheimer office. The camera rotates around a centre that is approximately the middle window bar which therefore appears nearly at the same position in each frame of the sequence. The sequence consists of 20 frames and three frames are shown in figure 6.10.

The inpaint region consists of seven separated black bars (each of size 200x10) in each frame of the sequence as depicted in the left image of figure 6.11. Over time the inpaint regions are fixed.

For this image sequence the apparent motion field is recovered using (6.1) and

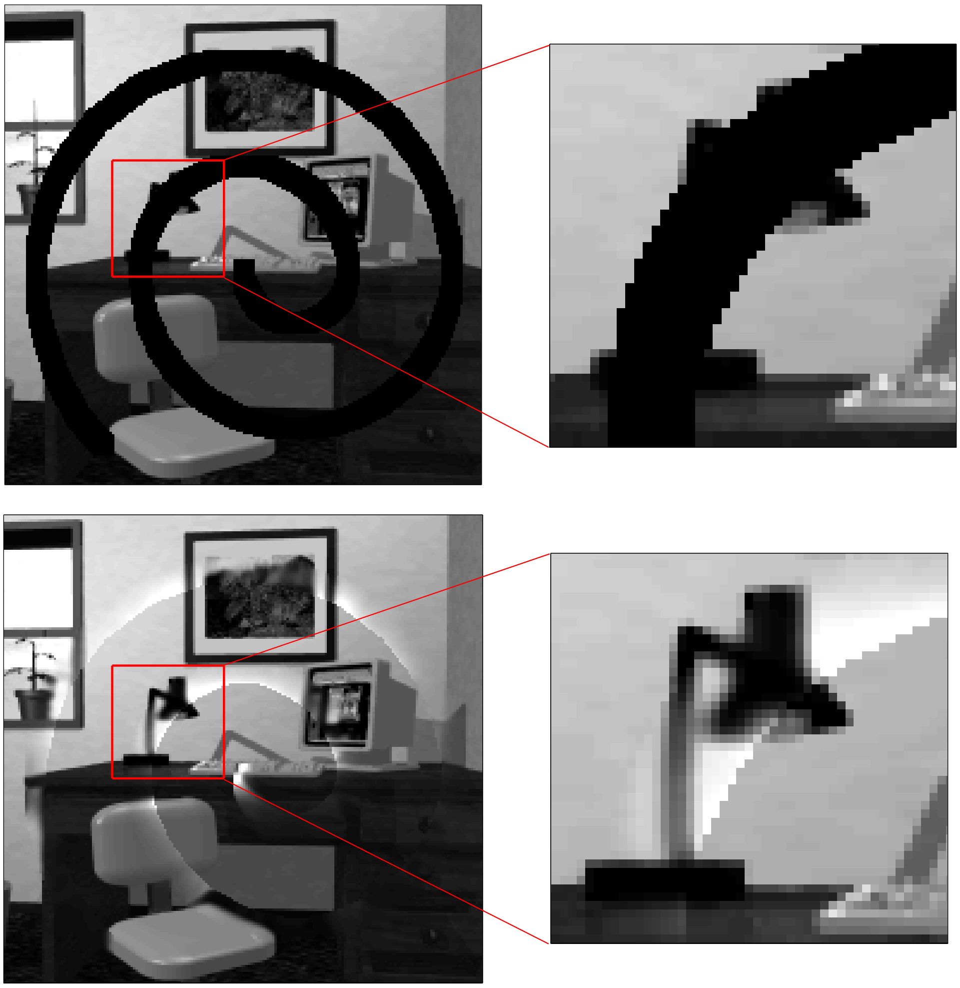


Figure 6.9: Structure preserving. **Top:** Image frame with missing image information (black spiral). Cutout: The image measurements for the desk light in this frame are nearly complete missing. **Bottom:** Result of our video inpainting approach. Due to the temporal image information transport, based on the Burgers term and advection-diffusion equation the information from previous images is transported over time. The structure of the object is well preserved.

(6.2) in the regions of missing image data. The inertia behaviour of the Burgers equation transports the vector field by itself into the region of missing image data (cf. experiment 3.4.3). The resulting velocity estimate is then used to transport image information into the inpaint region. The result is shown in the right image of figure 6.11 where we can observe that the structures of clouds, trees and the building are well reconstructed.

To show inpaint results for a larger inpaint region, we chose the same sequence now with an inpaint region that is large enough to cover the the middle window bar. It reaches a size of 30x150 pixels. In figure 6.12 we depict the inpaint results at three different times (frames).

Figure 6.10: The "Mannheim office" sequences. The camera rotates around the middle window bar which therefore stays nearly at the same position in each frame of the sequence. We depicted three image frames of this sequence.

Figure 6.11: The "Mannheim office" sequence consists of 20 frames. **Left:** Single frame of the "Mannheim office" sequence along with the inpaint region marked by black bars. **Right:** Video inpainting result for this frame of the sequence.

If the image sequence with respect to the motion field is long enough to inpaint the missing image data, we obtain good results for uniform motion patterns. This is no constraint for real applications as in general, local uniform motion fields are more common than non-uniform motions. This makes our approach applicable for other applications, like logo removal, or removal of disturbing objects in images, like the window bar in our experiment.

Figure 6.12: The "Mannheim office" sequence with a large inpaint region (size: 30x150 pixels). **Left:** Single frame of the "Mannheim office" sequence along with the inpaint region in the middle of the image. **Middle:** Video inpainting result for frame 4 of the sequence. **Right:** Inpaint result for the last frame.

6.4.4 Comparison

For a quantitative comparison we implemented the Navier-Stokes frame-by-frame inpainting approach from [10]. The inpainting is done using the method outlined in section 5.2.1 separately for each frame of the ”street” sequence (fig. 6.2). Using the ground truth image data the resulting root mean square (RMS) error plot for both approaches is shown in figure 6.13.

The conceptual difference between these two approaches lies in the way of regularisation. No information from previously or following image frames is used within the purely spatial approach from Bertalmío et al., as the inpainting is done individually frame-by-frame using the vorticity transport equation to transport image information from the surrounding area into the inpaint region. As a consequence, for each frame the entire inpainting process starts independently. In contrast our approach transports the velocity and image information over time. Therefore, knowledge from previous frames is used to progressively complete the inpaint area in the following frames. As observed in our previous experiments the transportation process over time is a more structure preserving technique (cf. fig. 6.14). The reconstructed image data in damaged regions is closer to the original image measurements. The spatial approach reconstructs the inpaint regions without exploiting knowledge about the scene structure from the previous frames: A smoother reconstruction (cf. fig. 6.14) is obtained as the isophote lines which reaches the inpaint region are smoothly continued from the exterior . Information is transported along these lines and the image measurements are updated by solving the Poisson equation, which leads to a continuous propagation of the inpaint region boundary and smoother image data in the inpaint regions.

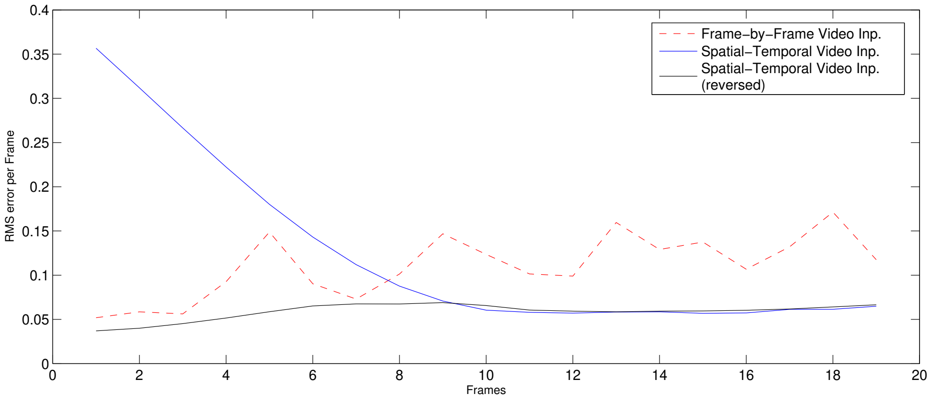


Figure 6.13: Comparison of our spatio-temporal video inpainting approach with the Navier-Stokes frame-by-frame video inpainting approach [10] using the ”street” sequence (fig. 6.2). The RMS-error is computed and plotted for each of the 19 frames of the sequence. Our two step dynamic video inpainting approach, which is based on spatio-temporal regularisation indicates a consistently lower error measurement as the purely spatial approach. Exploiting also the reverse inpainting our approach shows consistently a lower error.

Figure 6.14: Video inpainting result obtained by two different approaches: **Left:** Navier Stokes frame-by-frame approach. This approach has no knowledge about the scene structure from the previous frames. A smoother structure results from isophote lines which reaches the inpaint region from the exterior. Information is transported along these lines and the image measurements are updated by solving the Poisson equation. **Right:** Spatio-temporal video inpainting approach. The transportation process over time is a more structure preserving technique. Although not perfect reconstructed image data in damaged regions are closer to the original image measurements.

By comparing the two error plots, we clearly recognise this difference and observe that the error of the spatial approach shows a much higher variation than the temporal approach, which shows nearly a convergent evolution over time. Exploiting also the reverse inpainting our approach shows consistently a lower error. This indicates that our dynamic video inpainting approach which is based on spatio-temporal regularisation performs a reasonable video reconstruction.

6.5 Conclusion

The video inpainting algorithm we proposed in this chapter combines a physical transport mechanism with a distributed parameter approach for image motion estimation which incorporates dynamical prior knowledge. It results in a spatio-temporal sliding-window approach and is implemented using a two step algorithm and exploits approaches proposed in earlier chapters. The first ingredient – namely the distributed parameter approach – was subject of chapter 3 while the basic transportation process of the image data is based on the idea published in [9, 10].

Within our experiments we investigated uniform and non-uniform motion. The image information is transported from the flow field direction into the inpaint region. We like to point out that the inpainting process results in a structure preserving and reasonable result for uniform motion patterns. In the case of non-uniform motions it turned out that a good strategy is to exploit the reversed image sequence too, therefore transporting the image information from both flow directions into the inpaint region.

In contrast to spatial (frame-by-frame) inpaint approaches we incorporate temporal knowledge of the video sequence in the inpaint process using the motion information of the sequence. Due to the resulting velocity fields, which provides us with the transportation field for the image data our approach is able to transport the image information over time during the sequence. Structure information can - up to a certain degree - be preserved within the transportation process.

Recall that nowadays the number of frames per second (fps) that can be recorded by a camera, ranges from six frames per second for older cameras to 120 or more for professional high speed cameras. Therefore common image sequences have much more image frames than we used in our example sequences. This makes it very likely that enough information is provided in such sequences to result in good inpaint results even if the sequence is not reversed.

One aim that should be addressed in the future – and belongs to video editing – is the automatic removal of moving objects within an image sequence. This requires a reliable segmentation of the moving inpaint area and would mean that one has to be able to detect the motion boundaries quite accurately. This would help to increase the accurately of the computed motion vector fields.

Part III

Numerical Methods

7

Model Equations

In this chapter we are concerned in more detail with the particular partial differential equations which were used in connection with the approaches described in the previous chapters. Preferences and general notations will be introduced. We start with linear partial differential equations like the diffusion and advection equations. Then we move to non-linear equations where we are particularly interested in transport equations like the Burgers equation.

Propaedeutic literature to the field of partial differential equations can found in [47, 76, 136]. The books of Chorin and Marsden [32], Lax and Wendroff [86], Landau and Lifschitz [85], LeVeque [87], Toro [134], Thomas [130, 131] and Hundsdorfer [70] discuss partial differential equations in fluid dynamics.

7.1 Poisson Equation

The Poisson equation is one of the fundamental partial differential equations in physics and particularly one of the governing equations in fluid dynamics. The Poisson equation belongs to the class of second-order elliptic partial differential equations and yields in two dimensions with Dirichlet boundary conditions

$$\begin{aligned} \Delta s(x) &= \partial_{x_1}^2 s(x) + \partial_{x_2}^2 s(x) = \varrho(x)\ , && x \in \Omega, \\ s(x) &= g(x)\ , && x \in \partial\Omega, \end{aligned} \tag{7.1}$$

where $s = s(x)$, $x = (x_1, x_2)$, and Δ the Laplacian operator. In physics, s represents the density of a quantity and the second-order term Δs is responsible for the diffusion of s within the space region Ω. In physics the equation (7.1) is used to model steady incompressible inviscid flows, equilibrium temperature in a square slab or the static deformation of a membrane under distributed loads. Often one is interested in the steady state solution of these problems. The best known scenario is the steady state temperature distribution which is caused by a diffusion process with an appropriate heat source.

Another viewpoint is to think of the scalar quantity $s = s(x)$ as a potential field and of $\varrho(x)$ as its associated charge density, which induces the potential field. The boundary condition is given by $g(x)$. An example for a fixed boundary in a digital image is depicted in figure 7.1.

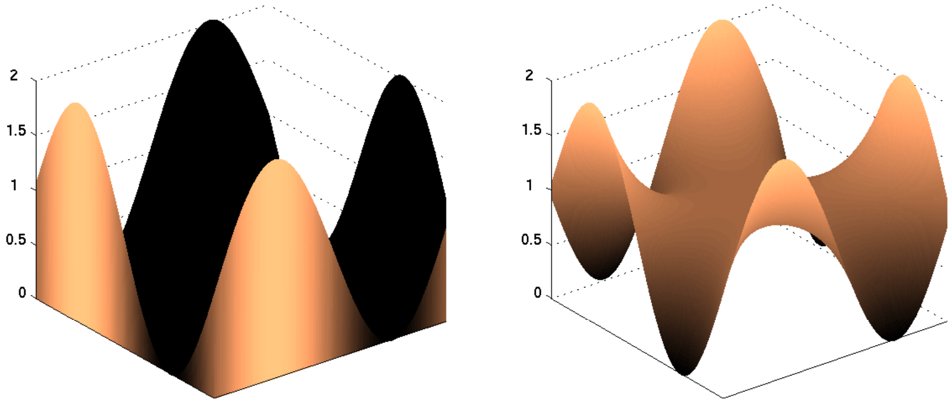


Figure 7.1: Computational result for the Poisson equation with fixed Dirichlet boundary conditions. **Left:** Input image ϱ with interior zero domain and fixed boundary values. **Right:** Result of the numerical solution by using the PCG method (sec. 9.1).

The Poisson equation serves as model problem in many textbooks on numerical analysis, see also section 8.1.5. We used the Poisson equation within our implementation of the Navier-Stokes inpainting approach of [9] (cf. sec. 5.2.1).

7.2 Diffusion Equation

The diffusion equation belongs – like the Poisson equation – to the class of second-order partial differential equations. But the diffusion equation is time-dependent and describes the diffusion process of a scalar quantity $s = s(x,t)$, $x = (x_1, x_2)$ over time $t \in [0,T]$ with respect to the kinematic viscosity coefficient $\nu > 0$:

$$\begin{aligned} \partial_t s(x,t) &= \nu\, \Delta s(x,t)\ , \quad & (x,t) \in \Omega \times [0,T], \\ s(x,t) &= 0\ , & (x,t) \in \partial\Omega. \end{aligned} \tag{7.2}$$

Viscosity is one of the central mechanical properties of fluids. It is a measure of the resistance of a fluid to deform under shear stress (i.e., a measure of fluid friction) and is perceived as the thickness of a fluid. Honey, for example – which is perceived quite thick – has a higher viscosity coefficient than water. Normally, fluids are resistant towards shear stress and thus are called viscous fluids. Fluids which has no resistance to shear stress are known as ideal fluids or inviscid fluids. We can distinguish between dynamic viscosity and kinematic viscosity. The dynamic viscosity term μ arises from the shear stress between two layers of a fluid. The shear stress τ between two layers is proportional to

original

$\nu = 0.5$

$\nu = 1$

$\nu = 2$

$\nu = 10$

$\nu = 100$

Figure 7.2: Diffusion process demonstrated at the well-known "Lena" image. The first image in the first row shows the original "Lena" image. The second image shows the computational result for the diffusion equation with a kinematic viscosity coefficient $\nu = 0.5$. The following images are computed with viscosity values $\nu = 1$, $\nu = 2$, $\nu = 10$ and $\nu = 100$. For further increasing values of ν all pixel intensity values of the image converge to the mean grey value of the image.

the velocity gradient $\partial_{x_2} v$ in the direction perpendicular to the relative motion of the layers. Formally this is expressed by the following relation:

$$\tau = \mu \partial_{x_2} v.$$

The kinematic viscosity ν is described by the ratio between the dynamic viscosity μ and the density $\rho = \rho(x, t)$ of the fluid. It is defined by

$$\nu = \frac{\mu}{\rho}.$$

Higher values of kinematic viscosity ν in equation (7.2) indicate a higher diffusion rate. The influence of ν on the diffusion process in images is depicted in figure 7.2. Equation (7.2) is also known as the general *heat equation*. The equation describes the time evolution of the density s of a quantity (in physics, e.g. heat, in computer vision applications often the grey values of the individual pixels). The numerical solution of the viscous diffusion equation is described in detail in section 8.1.6.

Figure 7.3: A anisotropic diffusion process applied to the ”Lena” image. **Left:** The original image. **Right:** The computational result for the anisotropic diffusion equation (7.3) with the parameter $\lambda = 2$.

7.2.1 Anisotropic Diffusion

A non-linear diffusion approach applied to images was first introduced by Perona and Malik [110] 1987. Weickert [141] applied the basic model of Perona and Malik amongst others to introduce regularisation techniques of this type of equations. They used a non-uniform process - known as anisotropic diffusion - that reduces diffusivity at edges and considered the scalar-valued case of a grey value image $I = I(x_1, x_2)$ as follows

$$\partial_t I = \nabla(g(|\nabla I|^2)\nabla I) \tag{7.3}$$

with the anisotropic diffusion term

$$g(|\nabla I|^2) = \frac{1}{1 + |\nabla I|^2/\lambda^2}\,, \quad \lambda > 0,$$

where $\nabla = (\partial_{x_1}, \partial_{x_2})^\top$ represents the nabla operator. Anisotropic diffusion allows for a directed diffusion process and has the aim in computer vision to prevent the diffusion over edges within an image while still allowing a diffusion along such edges. The effect of anisotropic edge enhanced diffusion is shown in figure 7.3.

7.3 Linear Advection Equation

The term advection refers to a transport of a conserved property like heat in a fluid. The linear advection equation is a first-order hyperbolic partial differential equation. Physically, this equation governs the motion of a conserved scalar quantity s which is advected or transported by a known velocity u. It can also be interpreted as the description of a wave with an amplitude s which propagates with a wave speed equal to u. In this thesis we consider the scalar advection equation

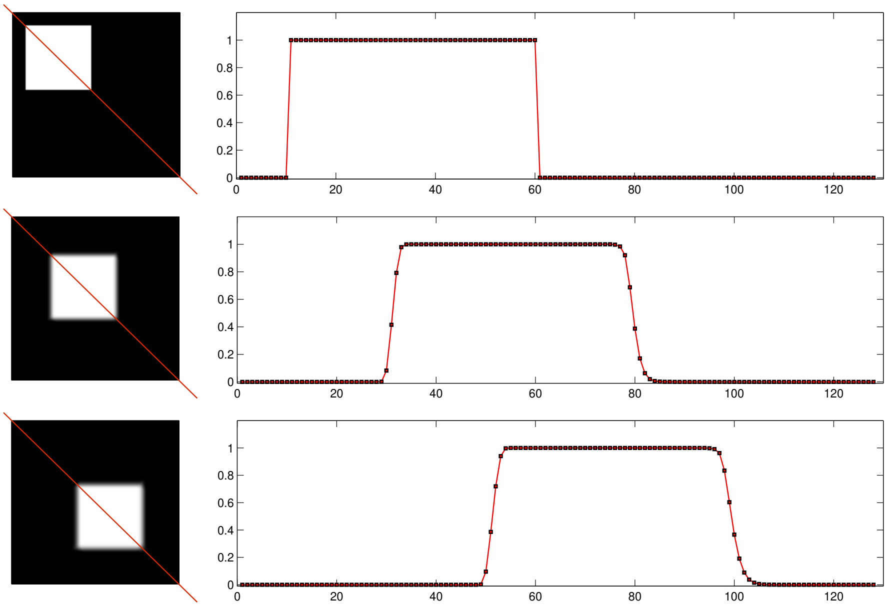


Figure 7.4: Numerical solution of the advection equation (7.4) for a small example transport problem. **Left column:** The white square is transported by a constant vector field u along the diagonal line from the upper-left edge to the down-right one. We show the transportation results at the times Δt, $\Delta t + 20$,and $\Delta t + 40$. **Right column:** Corresponding grey value evolution along the diagonal line indicated in the images on the left side.

$$\partial_t s + \nabla \cdot (su) = 0, \tag{7.4}$$

where $u = (u_1, u_2)^\top$, $u = u(x,t)$. We assume that we are concerned with incompressible fluids (or solenoidal velocity fields) which means that the fluid is divergence-free $\nabla \cdot u = 0$. Using these assumption we reformulate equation (7.4) as

$$\partial_t s + (u \cdot \nabla)s = 0. \tag{7.5}$$

For steady flows where $\partial_t s = 0$ is satisfied we obtain $(u \cdot \nabla)s = 0$ which means that the scalar quantity s is constant along a streamline.

In figure 7.5 we show a numerical solution of the advection equation (7.4) for a transportation problem. The first image in the left column of the figure shows a white patch which is transported by a constant vector field u along the diagonal from the upper-left corner to the lower-right one. The right column shows the evolution of the grey value along with the diagonal line indicated in the images of the left column. The white square is transported by our numerical implementation. Note, that the numerical schemes for the solution of hyperbolic systems have to cope with the formation of shocks. The details of our implementation for that is described in section 8.2.2.

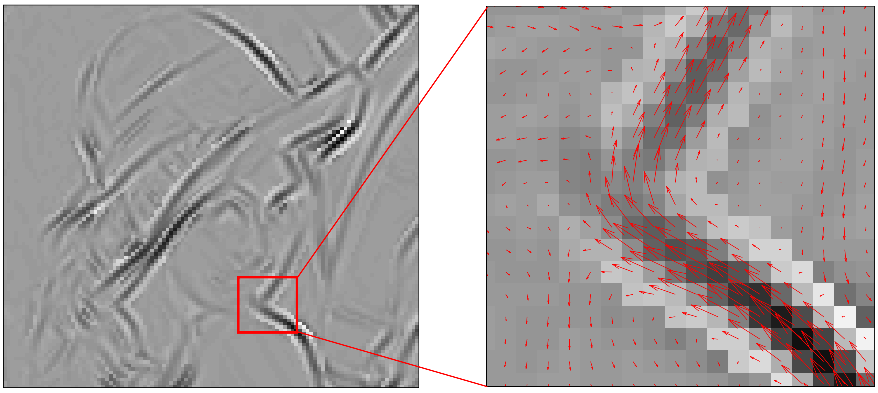


Figure 7.5: Numerical solution for a non-constant advection process. The "Lena" intensity image I is used to create a synthetically non-constant velocity field $u = (-\partial_y I, \partial_x I)^\top$. The initial image $I(x, 0)$ for this example was arbitrarily set homogeneously non-zero for each grey value. Then we used the synthetically velocity field u to transport this start image over 100 time steps. The resulting image $I(x, 100)$ is depicted on the left side of this figure. The right image shows a magnified part of the image along with the vector field u used for the transport.

To provide some details of the transportation process we show in figure 7.5 the numerical results for a non-constant advection process. For this example we artificially created a non-constant incompressible velocity field $u = (-\partial_y I, \partial_x I)^\top$ where I is a grey value image, in this case the "Lena" image. This field was then used to transport the grey values of a homogeneous coloured start image for 100 time steps. This process reveals some structural features of the image used to create the transportation field, but here we intended to show that the underlying transportation field can be of any shape.

7.4 Vorticity Transport Equation: Motion with Rotation

Incompressible flows are physically described by the Navier-Stokes equation. Recall that a flow is incompressible if and only if the velocity field u is divergence-free (i.e., $\nabla \cdot u = 0$) which means that there is no source or sink. The Navier-Stokes equation for incompressible flows reads

$$\partial_t u + (u \cdot \nabla)\, u = -\frac{1}{\rho} \nabla p + \nu \Delta u, \tag{7.6}$$

with von Neumann boundary conditions

$$\partial_n u = 0 \quad \text{on } \partial\Omega \tag{7.7}$$

along with the explicit constraint for incompressible fluids

$$\nabla \cdot u = 0. \tag{7.8}$$

Within these equations, n represents the unit vector in the direction outwards to the boundary $\partial\Omega$. The coefficient ν determines the kinematic viscosity, while ρ represents the density and $p = p(x,t)$ the scalar pressure of the fluid.

In the following we explain a reformulation of the Navier-Stokes equation (7.6) – for incompressible fluids – to an equation which depends only on the velocity u and no longer on the pressure p. To archive this we consider an important special case of the incompressible Navier-Stokes equation, the so-called *vorticity transport equation* (7.13). Note that this equation represents also a form of an advection-diffusion equation as shown in figure 7.6.

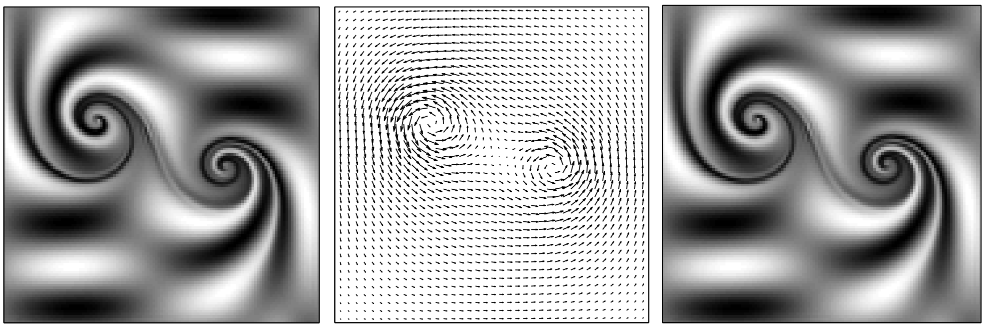


Figure 7.6: The vorticity transport equation, an advection-diffusion equation: We used as scalar vorticity initial value $\omega(x,0)$ a sinus image and transported this scalar quantity over time with a non-constant incompressible velocity field. **Left:** Scalar velocity quantity at time $t = 100$. **Middle:** Corresponding velocity field at time $t = 100$. **Right:** Scalar velocity quantity at time $t = 101$.

The *vorticity* $\omega = \omega(x,t)$ also known as a measure for the rotation of a velocity field is expressed in two dimensions [32] as:

$$\omega = \nabla \times u = \partial_{x_1} u_2 - \partial_{x_2} u_1. \tag{7.9}$$

It can be seen as the cross product between the gradient operator and the velocity field as it describes the rotation of the velocity field u.

The vorticity transport equation is obtained by applying the rotation operation $\nabla\times$ to equation (7.6) and by exploiting some vector identities. We start with a replacement of the term $(u \cdot \nabla)u$ in equation (7.6) by using the equation:

$$(u \cdot \nabla)u = \frac{1}{2}\nabla(u \cdot u) - u \times (\nabla \times u).$$

With that we obtain the following expression

$$\partial_t u + \frac{1}{2}\nabla(u \cdot u) - u \times (\nabla \times u) = -\frac{1}{\rho}\nabla p + \nu \Delta u.$$

By applying the rotation operator to this equation we get

$$\nabla \times \partial_t u + \frac{1}{2}\nabla \times \nabla(u \cdot u) - \nabla \times (u \times (\nabla \times u)) = -\frac{1}{\rho}\nabla \times \nabla p + \nu \nabla \times (\Delta u). \tag{7.10}$$

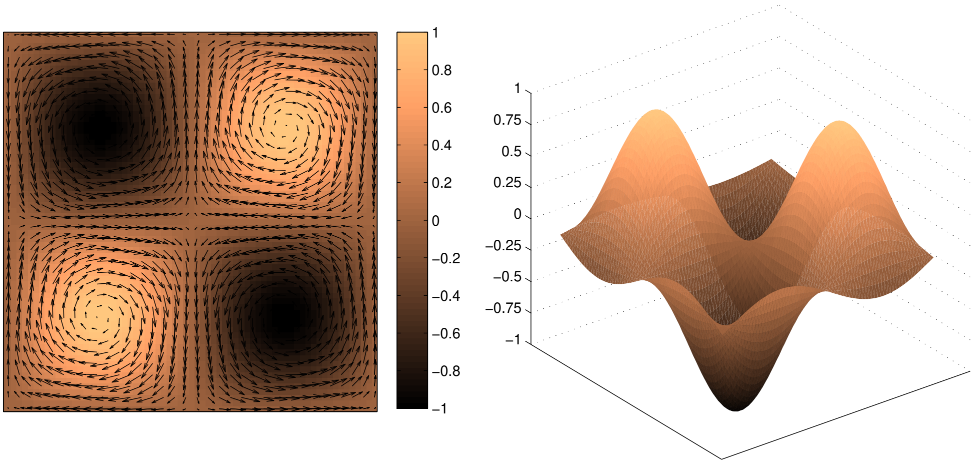


Figure 7.7: A synthetically example for the solution of the vorticity transport equation. **Left:** The scalar valued vorticity ω along with the corresponding velocity field u is shown. The vorticity values are in the range from -1 to $+1$. A positive vorticity value represent a clockwise rotation and negative values represent counter-clockwise rotations. The intensity of the colour indicates the strength of the rotation. **Right:** Surface plot of the scalar vorticity ω.

Note that in this expression the pressure and the second term of (7.10) vanishes due to the fact that for any scalar quantity ψ

$$\nabla \times \nabla\psi = 0$$

is valid. Therefore equation (7.10) is reduced to

$$\partial_t(\nabla \times u) - \nabla \times (u \times (\nabla \times u)) = \nu\nabla \times (\Delta u). \tag{7.11}$$

Using the fact that

$$\nabla \times (\Delta u) = \Delta(\nabla \times u)$$

and that the equality

$$\nabla \times (u \times \omega) = (\omega \cdot \nabla)u - (u \cdot \nabla)\omega + u(\nabla \cdot \omega) - \omega(\nabla \cdot u)$$

holds true we reformulate equation (7.11) as

$$\partial_t(\nabla \times u) - \nabla \times (u \times (\nabla \times u)) = \nu\nabla \times (\Delta u). \tag{7.12}$$

Finally, we obtain the desired vorticity transport equation

$$\partial_t\omega + (u \cdot \nabla)\omega = \nu\Delta\omega, \tag{7.13}$$

by using $\nabla \cdot \omega = 0$, $(\omega \cdot \nabla)u = 0$, and again exploiting the incompressibility condition (7.8).

Figure 7.7 illustrates an example for the solution of the vorticity transport equation. Figure 7.8 shows the vorticity transport equation applied to a particle flow image sequence. The resulting vorticity along with the corresponding velocity field u is depicted. For details of the implementation of the vorticity transport equation we refer to section 8.2.3.

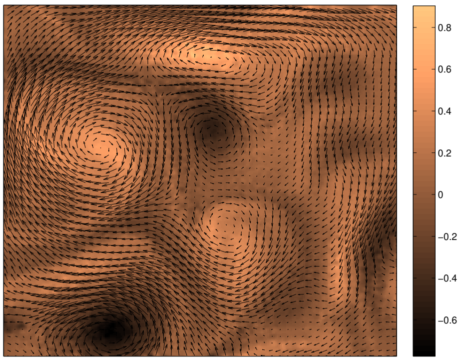


Figure 7.8: Particle flow example for the vorticity transport equation. The resulting vorticity along with the corresponding velocity field u is shown. Positive values stand for a clockwise rotation and negative vorticity values for a counter-clockwise rotation. The intensity of the pixel value correlates with the strength of the rotation.

7.5 Burgers Equation

In the following we are concerned with the Burgers equation which belongs to the class of hyperbolic systems of conservation laws. In physics the equation is used as a simplified model for turbulence, boundary layer behaviour, shock wave formation and mass transport. The Burgers equation has been studied for many decades and has been successfully applied to aero- and fluid dynamics [23, 145, 87, 49, 66].

The 2D Burgers equation is a non-linear partial differential of first-order

$$\partial_t u + (u \cdot \nabla) u = 0, \tag{7.14}$$

with initial data

$$u(x,t)_{t=0} = u_0. \tag{7.15}$$

The Burgers equation describes a transport process and is known as the advection term in many physical time-dependent dynamic equations. Therefore it is part of the most fundamental equation in fluid dynamics, e.g. the Navier-Stokes equation.

To get an idea of the dynamic behaviour of the Burgers equation, it is helpful to introduce first the material derivative which is a well-known quantity in fluid dynamics. The material derivative of u is the total derivative of the fluid velocity $u(x,t)$, $u = (u_1, u_2, u_3)^\top$ with respect to time t:

$$\begin{aligned} \frac{Du}{Dt} &= \partial_{x_1} u\, \dot{x}_1 + \partial_{x_2} u\, \dot{x}_2 + \partial_{x_3} u\, \dot{x}_3 + \partial_t u \\ &= \partial_t u + u_1\, \partial_{x_1} u + u_2\, \partial_{x_2} u + u_3\, \partial_{x_3} u \\ &= \partial_t u + (u \cdot \nabla) u, \end{aligned} \tag{7.16}$$

where $x(t)$ is the path followed by a fluid particle so that the velocity field is given by the components $\dot{x}_1 = u_1$, $\dot{x}_2 = u_2$, and $\dot{x}_3 = u_3$, which leads to

$u(x(t), t) = \frac{dx}{dt}(t)$.

The first part of equation (7.16), namely the time derivative, is known as the local derivative, which describes the physical change of the velocity at a fixed point with respect to time. The second part is known as the convective term of the equation. This non-linear term is also known as the inertia of the fluid that is involved in the transportation process. Physically, this is the rate of change due to the movement of a fluid element from one location to another in the flow field.

The material derivative $\frac{D}{Dt}$ describes the fact that the fluid is moving and that the positions of fluid particles change with time. The material derivative of u with respect to time is an expression for the acceleration which is observed within a fluid. In equation (7.14) the material derivative of u with respect to time is zero. This means that the acceleration of a fluid particle is zero over time and so the velocity of a fluid particle is constant over time.

Figure 7.9 illustrates the transportation process that is driven by the Burgers equation (7.14).

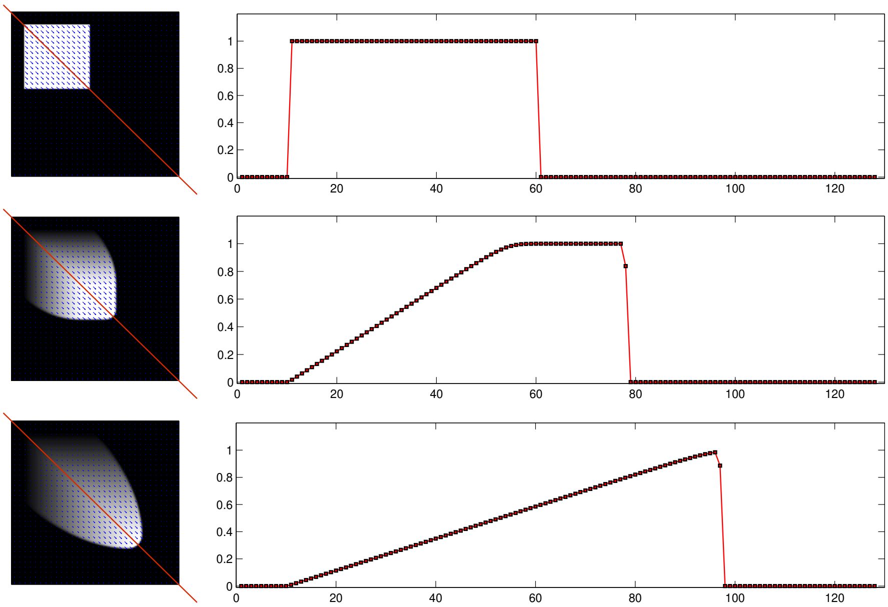


Figure 7.9: Solution of the Burgers equation (7.14): A velocity field transported by itself. The vector field is shown at times $t = 0, 20, 40$ and the vector magnitudes are represented by grey values. A shock front moves along the diagonal towards the lower-right corner in the images. In the absence of any further external information, a region of rarefaction arises due to mass conservation, acting like a short-time memory.

The quantity u is a 2D vector field which is – due to the Burgers equation – transported by itself. In the left column, the vector magnitudes are represented by grey values. The right column shows the time development of the field along the shown diagonal cut line in the image. We observe that a shock front is created which is transported towards the lower-right corner of the image.

The Burgers equation studied by Burgers himself [23] includes an additional viscous term $\nu \Delta u$:

$$\partial_t u + (u \cdot \nabla) u = \nu \Delta u. \tag{7.17}$$

This equation is a 2D model that includes the non-linear and viscous effects of fluid dynamics. The important point is that the viscosity term keeps the solution smooth for all time, preventing the breakdown of solutions that occur for the pure hyperbolic problem. Note in connection to this that equation (7.17) is no longer hyperbolic, as the additional viscous term makes it parabolic. The increasing relative weight of the viscous term prevents that the wave breaks and that discontinuities (shocks) emerge.

For small ν and smooth u the viscous term $\nu \Delta u$ is negligible and the solutions of equation (7.14) and (7.17) are nearly identical. If the solution of the PDEs in areas with smooth initial data along with a very small ν becomes discontinuous one speaks of shock formation or of a wave beginning to break. One can try to cope with such a situation by computing the weak solution of the PDE which becomes sharper and approaches the discontinuous solution for $\nu \to 0$. In order to deal with shock formations we use in our implementation an approximate solution of the Riemann problem. For details to the implementation of the Burgers equation (7.14) we refer to section 8.3.

8

Discretisation Techniques

The selection of a discretisation scheme for a mathematical formulation representing a specific problem is an inherent part of the solution process. The properties of the numerical computation are influenced for example by the *space-time discretisation* but also by the chosen *order of the numerical discretisation*.

The spacial discretisation refers to the choice of the mesh grid within the domain of interest Ω. The continuous space is thereby approximated by a finite number of grid points. Usually only for these points the numerical quantities are computed. The accuracy of the numerical approximation depends on the mesh size in relation to the actual continuous signal. Generally, the finer the space discretisation – therefore having narrower mesh points – the better is the numerical approximation scheme. The discretisation error tends to zero, when the mesh distance tends also to zero. The time discretisation plays an important role as well as this discretisation influences the stability of the numerical solution process. The stability conditions depend on the particular chosen numerical scheme (i.e. explicit or implicit schemes). The convergence properties of a numerical scheme is furthermore influenced by the order of the discretisation method.

After choosing an appropriate mesh grid the model equations can be discretised. Therefore, differential equations or integral formulations must be transformed to discrete algebraic structures, where the quantities of the unknown variables are defined on the mesh points.

In this thesis we are mainly concerned with *time dependent* formulations of problems. Naturally these partial differential equations involve the time derivative. The discretisation of time derivatives generate a system of equations for the unknowns, which are given by functions at the actually time step and the previous time steps. One distinguish between *explicit* and *implicit* numerical schemes. In explicit approaches the computation of the current value of an unknown variable depend only on the values already computed for previous time steps. In *implicit* schemes the new value at the actual time step depends also on values of the actual time step. To compute such systems of equations there exists several types of numerical solvers, some of them are discussed in chapter 9.

In the following we present the two most common techniques for a space-time discretisation of differential operators: *finite difference* and *finite element* methods. Fundamental literature in this sphere of action include the works of Colella and Puckett [36], Hirsch [66], Lax and Wendroff [86], LeVeque [87], Toro [134] and Thomas [130]. These techniques provide the fundamentals to obtain a discretisation for a partial differential problem formulation.

8.1 Finite Differences

The foundations for finite difference methods can be found in the *definition of derivatives* and the approximation framework given by *Taylor expansions*. First of all, we define the underlying mesh grid design.

8.1.1 Fundamentals

The computation domain Ω is divided into cells. We discretise the space-time $(x_1, x_2, t)=(x, t)$ volume using a mesh with width Δx and a time discretisation step Δt. In this thesis we consider only regular or *uniform* discretisation meshes.

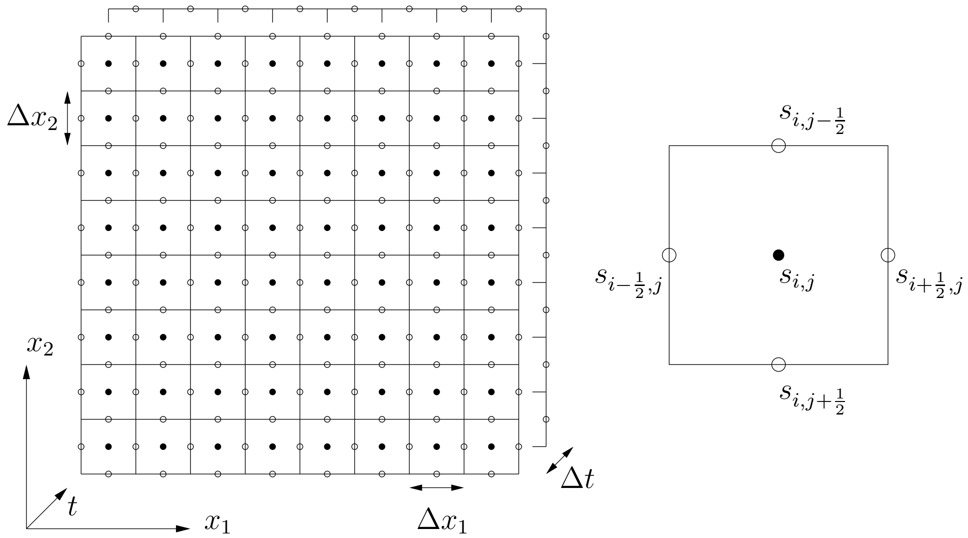


Figure 8.1: Space-time discretisation of the domain Ω. **Left:** The domain is divided into same sized cells. The spacial centre of the cells along with their boundaries is indicated as well. **Right:** Magnification of a single cell from the grid on the left side. The quantity s is defined on the cell centre (i, j). Boundaries belonging to the cell are indicated by appropriate half-indices.

Here the mesh is chosen to be uniform which means that Δx and Δt are constant. Such a discretisation scheme is depicted in figure 8.1.

The variable $s^n_{i,j}$ refers the quantity at the position $x = (x_1, x_2) = (i\Delta x_1, j\Delta x_2) =: (i, j)$ at time $t = n\Delta t =: t^n$ and is assigned to the centre of the cell. Numerical quantities with half-indices refer to the cell boundaries. For example $s^n_{i+\frac{1}{2},j}$ refers to the boundary between the cell (i, j) and the cell $(i + 1, j)$ at time $t = n\Delta t$ (cf. fig. 8.1 right).

The discretised quantities $s^n_{i,j}$ on the grid should approximate the exact values $s(x, t)$ of the continuum. A particular grid value $s^n_{i,j}$ can be seen as an approximation for the average value of the real function $s(x, t)$ within the appropriate cell boundary

$$s^n_{i,j} = \frac{1}{\Delta x_1}\frac{1}{\Delta x_2}\int\limits_{(j-\frac{1}{2})\Delta x_2}^{(j+\frac{1}{2})\Delta x_2}\int\limits_{(i-\frac{1}{2})\Delta x_1}^{(i+\frac{1}{2})\Delta x_1} s(x_1, x_2, t^n)dx_1dx_2. \tag{8.1}$$

The integral form of conservation laws describes in fact the time evolution of such integrals. (Note, in the following we denote partial derivatives ∂_x by $\frac{\partial}{\partial x}$).

The idea of finite differences has its origin in the definition of the derivative of a function $s(x)$ at a point x:

$$\frac{\partial s}{\partial x} = \lim_{\Delta x \to 0} \frac{s(x + \Delta x) - s(x)}{\Delta x}. \tag{8.2}$$

The right hand side of equation (8.2) is an approximation for the derivative $\frac{\partial s}{\partial x}$ for an infinitesimal Δx. If Δx is a finite value the derivative is approximated up to an error which is known as the *local truncation error* of the discretisation scheme. The approximation becomes more accurate with decreasing mesh size, with $\Delta x \to 0$.

Furthermore, the accuracy of a particular approximation scheme depends on the order to which the Taylor expansion is computed. The highest power of Δx that occurs in the scheme defines the *order of the difference approximation* and also measures how fast the error tends to zero. The Taylor expansion

$$s(x + \Delta x) = s(x) + \Delta x\, \frac{\partial s}{\partial x} + \mathcal{O}(\Delta x^2) \tag{8.3}$$

includes only a first-order term. The higher-order terms are denoted as error term $\mathcal{O}(\Delta x^2)$. The reformulation of equation (8.3) results in a finite difference approximation with an error of first-order:

$$\frac{s(x + \Delta x) - s(x)}{\Delta x} = \frac{\partial s}{\partial x} + \mathcal{O}(\Delta x). \tag{8.4}$$

Comparing equation (8.4) with equation (8.2) we observe that it represents a first-order finite difference approximation for the derivative $\frac{\partial s}{\partial x}$. Other finite difference approximations for the derivative (8.2) are shortly summarised in section 8.1.4.

In the following we are concerned with convergence conditions for finite difference schemes.

8.1.2 Consistency, Stability and Convergence

In this section we shortly repeat necessary and sufficient conditions for convergence of linear finite difference methods. As already mentioned the local truncation error represents a measure for the accuracy of a numerical discretisation scheme. Here, we denote this error as ϵ (with appropriate subscripts) and it can be obtained by comparing the exact solution of a partial differential equation with the finite difference approximation of the partial differential equation.

To introduce the local truncation error [36], we make use of the following notations: First we denote with L a linear discrete evolution operator that evolves the states according to the discretised partial differential equation. The term s_e^n refers to the exact solution of the PDE at the discrete points at the time n. With that one defines the *local truncation error* of a finite difference scheme as follows:

$$\epsilon = s_e^{n+1} - L s_e^n.$$

This represents a measures how much the discrete evolution differs from the exact solution after one time step. A numerical scheme is said to be *consistent* when the local truncation error tends to zero when the step size Δx and Δt tends to zero:

$$\|\epsilon_{i,j}^n\| \to 0 \quad \text{as } \Delta x \to 0, \Delta t \to 0. \tag{8.5}$$

The *global error* of the numerical scheme is of order l if

$$\|\epsilon_{i,j}^n\| = \mathcal{O}(\Delta t \sum_{p+q=l} \Delta x^p \Delta t^q)$$

for all sufficiently smooth initial data with compact support, where $p, q \geq 0$ and $p + q = l$.

A finite difference method is known to be *stable* if the equation

$$\|L s_e^n - L s^n\| \leq C \|s_e^n - s^n\| \tag{8.6}$$

can be fulfilled, where the constant C is independent of the exact solution s_e^n and the approximated solution s^n of the PDE [36].

The definitions of stability and consistency lead to the following theorem:

Lax Equivalence Theorem:

> The conditions of a consistent discretisation scheme (8.5) and the stability of a linear finite difference method (8.6) is the necessary and sufficient condition for *convergence*.

This is known as the fundamental *convergence theorem* for linear finite difference methods [86, 36] and indicates that for a consistent, linear method, stability is necessary and sufficient for convergence.

8.1.3 The CFL-Condition

Some concepts of convergence and stability of finite difference schemes were published by Courant, Friedrich and Lewy [37] in 1928. Among other things they presented their well-known Courant-Friedrich-Lewy (CFL) condition. It represents a fundamental stability condition for explicit finite difference schemes:

$$0 < \sigma_{CFL} = a\frac{\Delta t}{\Delta x} \leq 1. \tag{8.7}$$

The so called Courant number σ_{CFL} is a dimensionless quantity, which depends on the speed a, the grid space Δx and the time space Δt. This condition play a fundamental role in the computation of wave and convection equations. It indicates that the distance between two mesh points must be larger than the distance that can be reached by a disturbance that propagates with the speed a during the time interval Δt.

Disturbances which propagate with the velocity a are known to have the characteristic speed $a = \frac{dx}{dt}$. The characteristics define the domains in which states of the differential equation can have a mutually influence on each other. Therefore the CFL-condition (8.7) means that the domain of mutually influence in time direction should be contained in the spacial domain of mutually influence of the discretised equations, which are defined on the grid space Δx.

The Courant number σ_{CFL} can also be understand as the ratio of two speeds, namely the wave speed a and the grid speed $\frac{\Delta x}{\Delta t}$. The wave speed a is defined by the initial conditions, the grid speed depends on the desired accuracy of the numerical discretisation scheme. Apparently the only term that can be chosen is the time step Δt. But this choice is restricted as one has to take into consideration the stability conditions and the particular discretisation schemes of interest.

8.1.4 Standard Finite Difference Formulas

In this section we provide three finite difference formulas that can be used to numerically approximate a derivation. Let us consider a two-dimensional space as shown in figure 8.1. Again, $s_{i,j}$ denotes the value of the function $s(x)$ at the discretised position (i,j). The finite difference approximation for the first derivative of the quantity $s_{i,j}(x)$ can be defined by one of the following three finite difference formulas.

Forward Difference

The first-order forward difference is given by

$$\left(\frac{\partial s}{\partial x_1}\right)_{i,j} = \frac{s_{i+1,j} - s_{i,j}}{\Delta x_1} + \mathcal{O}(\Delta x_1). \tag{8.8}$$

Backward Difference

The first-order backward difference is given by

$$\left(\frac{\partial s}{\partial x_1}\right)_{i,j} = \frac{s_{i,j} - s_{i-1,j}}{\Delta x_1} + \mathcal{O}(\Delta x_1).$$

Central Difference

The second-order central difference is given by

$$\left(\frac{\partial s}{\partial x_1}\right)_{i,j} = \frac{s_{i+1,j} - s_{i-1,j}}{2\Delta x_1} + \mathcal{O}(\Delta x_1^2).$$

In our implementations we are mainly concerned with these difference schemes.

8.1.5 Stationary Problems

Discretisation of the 2D Poisson equation

In this section we are concerned with the discretisation of the two-dimensional Poisson equation (7.1) that we used in our implementations (cf. sec. 7.1):

$$\begin{aligned} \Delta s &= \phi \quad \text{in}\Omega \\ s &= g \quad \text{on}\partial\Omega. \end{aligned} \tag{8.9}$$

The aim is to compute the approximation of the quantities $s = s(x)$ and $\phi = \phi(x)$, $x = (x_1, x_2)$ at the discretised cell centres on a two-dimensional squared grid within the domain Ω using an equal mesh spacing Δx as depicted in figure 8.2.

This problem is stated with Dirichlet boundary conditions as $s = g$ on the boundary $\partial\Omega$. The pixel position (i,j), $i = 1, ..., M$, $j = 1, ..., N$ refers to spacial x_1 and x_2 position $i\Delta x_1$ and $j\Delta x_2$, respectively. In order to discretise the left hand side of equation (8.9)

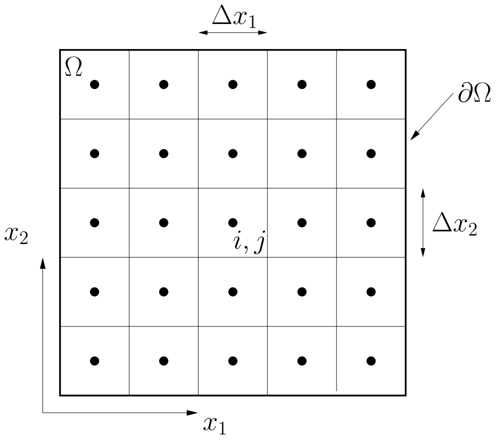


Figure 8.2: Discretisation domain Ω of the 2D Poisson equation with boundary region $\partial\Omega$.

$$\Delta s = \frac{\partial^2 s}{\partial x_1^2} + \frac{\partial^2 s}{\partial x_2^2}$$

we use a combined forward-backward discretisation

$$(\Delta s)_{i,j} = \frac{s_{i+1,j} - 2s_{i,j} + s_{i-1,j}}{\Delta x_1^2} + \frac{s_{i,j+1} - 2s_{i,j} + s_{i,j-1}}{\Delta x_2^2}.$$

This corresponds - using the grid size $\Delta x_1 = \Delta x_2 = 1$ - to the standard five-point Laplacian:

$$(\Delta s)_{i,j} = s_{i+1,j} + s_{i-1,j} - 4s_{i,j} + s_{i,j+1} + s_{i,j-1}. \tag{8.10}$$

Using equation (8.10) we express the discretisation within the interior domain Ω as:

$$(\Delta s)_{i,j} = \phi_{i,j}.$$

For the Dirichlet boundary conditions $s = g on \partial\Omega$ we have to enforce original (image) values g on the boundary $\partial\Omega$ as for example depicted in figure 7.1. Putting the pixel values into a vector s and keeping track of the actual boundary pixels the problem transforms into a linear equation system [36]. With the matrix

$$A_{ij} = -4\delta(i,j) + \delta(i,j+1) + \delta(i,j-1) + \delta(i,j+(n+1)) + \delta(i,j-(n+1)),$$

where

$$\delta(i,j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

the problem is stated as the following equation system

$$\begin{aligned} As &= \phi \\ s &= A^{-1}\phi. \end{aligned}$$

Such a system can be solved using iterative or direct solvers and multigrid methods. Some of these methods are described in section 9.1 and 9.3. In figure 7.1 we presented an example for a computational result of the Poisson equation where we used fixed Dirichlet boundary conditions.

8.1.6 Crank-Nicholson Scheme

The Crank-Nicholson scheme is an implicit finite differencing scheme. The scheme is unconditionally stable and second-order accurate in space x and time t. Compared to a first-order scheme, one can accomplish a given level of accuracy with a coarser grid and hence with less costs for the computation.

We study the Crank-Nicholson scheme at the scalar diffusion equation (7.2) in section 7.2 for two space dimensions $x = (x_1, x_2)$

$$\frac{\partial s}{\partial t} = \nu \Delta s, \tag{8.11}$$

where ν represents the kinematic viscosity coefficient and $s = s(x)$ represents a scalar quantity. Using the standard five-point Laplacian the discretisation of equation (8.11) results in

$$\begin{aligned} \frac{s^{n+1} - s^n}{\Delta t} &= \nu\, \Delta_{i,j} \Big(\frac{s^n + s^{n+1}}{2} \Big) \\ \left(I - \frac{1}{2} \Delta t \nu\, \Delta_{i,j} \right) s^{n+1} &= \left(I + \frac{1}{2} \Delta t \nu\, \Delta_{i,j} \right) s^n. \end{aligned}$$

This results in a linear system

$$\mathcal{L}\, \tilde{s} = \varrho, \tag{8.12}$$

where $\tilde{s}$ represents the quantity at time level $n+1$ as depicted in figure 8.3 and ϱ on the right-hand side is given by

$$\varrho = \left(I + \frac{1}{2} \Delta t \nu\, \Delta_{i,j} \right) s^n. \tag{8.13}$$

The linear operator $\mathcal{L}$ is defined as

$$\mathcal{L} = \left(I - \frac{1}{2} \Delta t \nu\, \Delta_{i,j} \right). \tag{8.14}$$

We write the linear operator in a matrix form on s as

$$(\mathcal{L}\,\tilde{s})_{i,j} = \left(I + \frac{4\Delta t\,\nu}{2h^2}\right)\tilde{s}_{i,j} - \frac{\Delta t\,\nu}{2h^2}(\tilde{s}_{i+1,j} + \tilde{s}_{i-1,j} + \tilde{s}_{i,j+1} + \tilde{s}_{i,j-1}). \quad (8.15)$$

The resulting linear system with Dirichlet boundary conditions can be solved with an appropriate solver, like PCG or multigrid solver (cf. chap. 9).

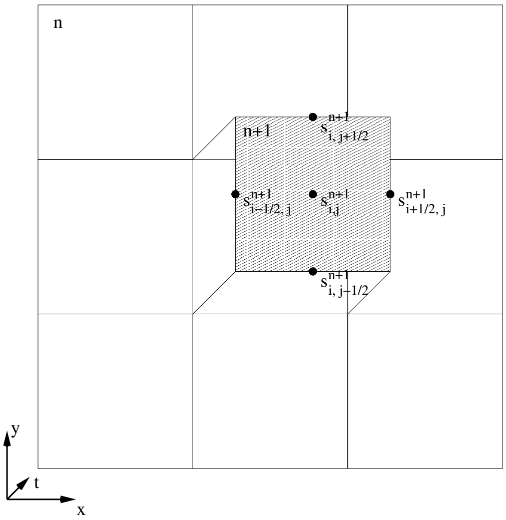


Figure 8.3: Illustration of the time-space discretisation

8.2 Conservative Finite Difference Method

A particular important class of partial differential equations are equations, which model conservation laws in physics. There is a high desire for algorithms that are able to imitate the correct behaviour for solutions to conservation laws when discontinuities such as shock waves are formed. The work of Hou and Le Floch [69] has shown that non-conservative schemes are not able to converge to the correct solution if a shock wave is present in the solution. But for conservative numerical methods Lax and Wendroff [86] found that this methods, if convergent, do converge to a weak solution of the conservation law. As we are working with equations that can generate shocks we decided to use conservative methods. Alternative schemes that are also shock fitting can be found in [95, 94] and adaptive conservative schemes are the topic of [133, 78].

Conservative Method:
A conservative scheme for the two-dimensional conservation law reads

$$\frac{\partial s}{\partial t} + \nabla \cdot F(s) = 0, \quad (8.16)$$

where s represents the conserved quantity and $F(s)$ denotes the physical flux. It can be discretised in the following form

$$s_{i,j}^{n+1} = s_{i,j}^{n} + \frac{\Delta t}{\Delta x_1}\left(F_{i-\frac{1}{2},j} - F_{i+\frac{1}{2},j}\right) + \frac{\Delta t}{\Delta x_2}\left(F_{i,j-\frac{1}{2}} - F_{i,j+\frac{1}{2}}\right), \tag{8.17}$$

where $s = s(x,t)$, $s = (s_1, s_2)^\top \in \mathbb{R}^2$, $x = (x_1, x_2) \in \mathbb{R}^2$, $t \in \mathbb{R}$ and $F \in \mathbb{R}^2$.

$$F_{i+\frac{1}{2},j} = F_{i+\frac{1}{2},j}(s_{i-1,j}^{n}, ..., s_{i+1,j}^{n}),$$

represents the numerical flux through the cell edge $(i+\frac{1}{2}, j)$ as indicated in figure 8.4. Note that it depends on $s_{i-1,j}^n, s_{i,j}^n$ and $s_{i+1,j}^n, s_{i,j}$ in the cell and neighbouring cells. The values F are the discrete approximations of the continuous physical flux $F(s)$ in the conservation law (8.16).

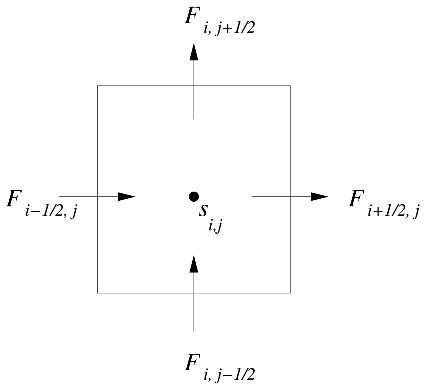


Figure 8.4: Numerical fluxes in and out of the cell (i,j).

In the following we summarise some numerical details for linear and non-linear conservation laws, which we used in this thesis. In particular we describe the discretisation schemes for the advection, vorticity transport and Burgers equation.

8.2.1 Geometric Limiters - Flux Corrected Transport

Flux limiters are used in high-resolution schemes to avoid the spurious oscillations that would otherwise occur with high-order spatial discretisation schemes due to sharp changes like shocks or discontinuities in the solution domain.

The basic idea of a geometric limiter for finite difference methods is, to apply a limiter to an interpolation function $u_I(x)$, and then use an upwind method to calculate the fluxes and advance the solution. This implies that no overshoot or undershoot can occur in the solution and oscillations at discontinuities are avoided.

In the following we describe the limiter construction in Van Leer's scheme. Van Leer's scheme combines Fromm's upwind method [50], which is described in the following section with the geometric limiter of Van Leer [137]. The interpolation function $u_I(x)$ to the exact solution $u(x)$ in Fromm's Method is defined by

$$u_I(x) = u_j + \frac{x - j\Delta x}{\Delta x}\,\Delta u_j.$$

Van Leer's scheme adjusts the slopes Δu_j

$$\Delta u_j = \frac{\partial u}{\partial x}\Big|_{j\Delta x} = \frac{u^n_{j+1} - u^n_{j-1}}{2}$$

in Fromm's method. These slopes are adjustments such that the interpolation function $u_I((j+\frac{1}{2})\Delta x)$ lies in the range between u^n_{j+1} and u^n_{j-1} as depicted in figure 8.5.

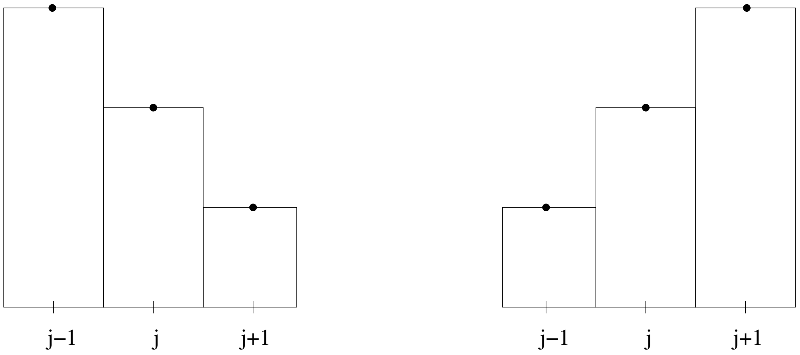


Figure 8.5: Left: The state u^n_{j-1} is the maximum value at time n in the neighbourhood $\{j-1, j, j+1\}$. **Right:** The state u^n_{j+1} is the maximum value at time n in the neighbourhood $\{j-1, j, j+1\}$.

In the following we are concerned with the question, when the interpolation function $u_I((j+\frac{1}{2})\Delta x)$ stays in the desired range and when a geometric limiter should be applied to force it to stay in the range.

We want to avoid over- or undershoots. That means, that new extreme values should not occur. The problem is that it is possible for two fluxes to cause a value to exceed the maximum and minimum limits, although each flux – acting separately – might remain in the bounds. Such a situation occur if φ the quantity, which defines the slope direction is positive:

$$\varphi = (u^n_{j+1} - u^n_j)(u^n_j - u^n_{j-1}) \;>\; 0. \tag{8.18}$$

Two cases lead to a positive value for φ. Firstly, if both differences of (8.18) are negative:

This situation is depicted in the left image of figure 8.5 where the value u^n_{j-1} is the maximum and u^n_{j+1} the minimum value with respect to the centred value j. In the shown case the interpolation function should clearly stay within the range between these two values.

Secondly, if both differences are positive:

In the right image of figure 8.5 the value u^n_{j+1} is the maximum and u^n_{j-1} the minimum value. Again, the interpolation function should stay in the range between these two values. The Fromm interpolation function

$$\begin{aligned} u_I\Big(\Big(j+\frac{1}{2}\Big)\Delta x\Big) &= u_j + \frac{((j+\frac{1}{2})\Delta x) - j\Delta x}{\Delta x}\,\Delta u_j \\ &= u_j + \frac{1}{2}\,\Delta u_j \end{aligned}$$

can attain the maximum and minimum value u^n_{j+1} and u^n_{j-1}, respectively. The value u^n_{j+1} is reached

$$u_I\Big(\Big(j+\frac{1}{2}\Big)\Delta x\Big) = u^n_{j+1}$$

when the slope $\Delta u_j = 2(u^n_{j+1} - u^n_j)$ is used:

$$u_I\Big(\Big(j+\frac{1}{2}\Big)\Delta x\Big) = u_j + \frac{1}{2}2(u^n_{j+1} - u^n_j).$$

The value u^n_{j-1} is obtained

$$u_I((j+\frac{1}{2})\Delta x) = u^n_{j-1}$$

when the slope $\Delta u_j = 2(u^n_j - u^n_{j-1})$:

$$u_I\Big(\Big(j+\frac{1}{2}\Big)\Delta x\Big) = u_j - \frac{1}{2}2(u^n_j - u^n_{j-1}).$$

If $\varphi > 0$ we ensure that the interpolation function stays within the defined range by introducing the Van Leer slope $\Delta^{VL}u_j$:

$$\Delta^{VL}u_j = \begin{cases} S_j \cdot \min\{2|u^n_{j+1} - u^n_j|, \frac{1}{2}|u^n_{j+1} - u^n_{j-1}|, 2|u^n_j - u^n_{j-1}|\} & \text{if} \quad \varphi > 0, \\ 0 & \text{if} \quad \varphi < 0. \end{cases}$$

Here S_j is defined by

$$S_j = sign(u^n_{j+1} - u^n_{j-1}).$$

This slope prevents the construction of new minima or maxima within the solution and prevents therefore undesired instabilities in form of oscillations. In the following section we show how to combine Fromm's Method with the geometric limiter of Van Leer.

8.2.2 Fromm's Second-Order Upwind Method

In the following we use Fromm's upwind method to compute the incompressible advection equation in two dimensions. It represents a second-order finite-difference method. In two dimensions, the model advection equation is defined as

$$\frac{\partial s}{\partial t} + (u \cdot \nabla) s = 0, \tag{8.19}$$

with the advective quantity $s = s(x,t)$, $x = (x_1, x_2)$ and the velocity field $u = (u_1, u_2)^\top$. Using the divergence-free condition $\nabla \cdot u = 0$ we obtain the following identity:

$$\begin{aligned} \nabla \cdot (u\, s) &= (\nabla \cdot u) s + u \cdot (\nabla\, s) \\ &= (u \cdot \nabla) s. \end{aligned} \tag{8.20}$$

Using (8.20) we reformulate equation (8.19) into

$$\frac{\partial s}{\partial t} + \nabla \cdot (u\, s) = 0. \tag{8.21}$$

To discretise this equation we employ Fromm's finite difference upwind scheme in combination with Van Leer's limiter. This results in an explicit scheme for the computation of the quantity $s_{i,j}^{n+1}$ at the next time step t^{n+1}. It is assumed that $s_{i,j}^n$ is known which is supposed to be the average value of the function $s(x,t)$ over the cell (i,j) at time t^n. The conservation form (such as in (8.17)) in the Fromm formulation turns out to be

$$s_{i,j}^{n+1} = s_{i,j}^n + u_1 \frac{\Delta t}{\Delta x_1} \left(s_{i-\frac{1}{2},j} - s_{i+\frac{1}{2},j} \right) + u_2 \frac{\Delta t}{\Delta x_2} \left(s_{i,j-\frac{1}{2}} - s_{i,j+\frac{1}{2}} \right), \tag{8.22}$$

where exemplarily the flux over the left cell boundary is given by

$$F_{i+\frac{1}{2},j} \;=\; u_{1_{i+\frac{1}{2},j}} \Delta t\, s_{i+\frac{1}{2},j}^{n+\frac{1}{2}}.$$

The flux over the other boundaries obtained by the analogous computation. We implemented a predictor-corrector scheme with second-order accuracy. The computation of the quantities $s_{i-\frac{1}{2},j}$, $s_{i+\frac{1}{2},j}$, $s_{i,j-\frac{1}{2}}$ and $s_{i,j+\frac{1}{2}}$ at half steps represent the *predictor step* in that scheme. The conservative differencing (8.22) is then the *corrector step*. The second-order accuracy is archived by using an approximation with local truncation error $\mathcal{O}(\Delta t^2)$. The scheme turns out to be second-order accurate in smooth regions of the solution and first-order accurate near to discontinuities.

In the following we describe the predictor step of the method to compute the half-step values at the cell boundaries. We describe the computation only for the single value $s_{i+\frac{1}{2},j}$. But the other quantities are obtained analogously.

Predictor Step

In order to find an approximation for the half-step value $s_{i+\frac{1}{2},j}$ (that lies at the right cell boundary) we use the Taylor expansion and get the following expression:

$$s^{n+\frac{1}{2}}_{i+\frac{1}{2},j} = s^n_{i,j} + \frac{1}{2}\Delta x_1 \frac{\partial s}{\partial x_1}\Big|_{i\Delta x_1, j\Delta x_2} + \frac{1}{2}\Delta t \frac{\partial s}{\partial t}\Big|_{n\Delta t} + \mathcal{O}(\Delta x_1^2, \Delta t^2). \tag{8.23}$$

Using then the advection equation (8.21) – which exploits the incompressible flow constraint – to substitute the temporal derivative one obtains:

$$\begin{aligned} s^{n+\frac{1}{2}}_{i+\frac{1}{2},j} &= s^n_{i,j} + \frac{1}{2}\Delta x_1 \frac{\partial s}{\partial x_1}\Big|_{(i,j)} + \frac{1}{2}\Delta t \left(-u_1 \frac{\partial s}{\partial x_1} - s\frac{\partial u_1}{\partial x_1} - \frac{\partial (u_2 s)}{\partial x_2} \right)\Big|_{(i,j)} \\ &= s^n_{i,j} + \frac{1}{2}\left(1 - u_1 \frac{\Delta t}{\Delta x_1}\right) \Delta x_1 \frac{\partial s}{\partial x_1}\Big|_{(i,j)} - \frac{1}{2}\Delta t\; s\, \frac{\partial u_1}{\partial x_2}\Big|_{(i,j)} \\ &\quad - \frac{1}{2}\Delta t \frac{\partial (u_2 s)}{\partial x_2}\Big|_{(i,j)}. \end{aligned} \tag{8.24}$$

Note that the error terms of equation (8.23) were just dropped. To discretise the last term of (8.24) we use the second term of the finite-difference approximation

$$\nabla\,(u\;s)^{n+\frac{1}{2}}_{i,j} = \frac{u_{1_{i+\frac{1}{2},j}} s^{n+\frac{1}{2}}_{i+\frac{1}{2},j} - u_{1_{i-\frac{1}{2},j}} s^{n+\frac{1}{2}}_{i-\frac{1}{2},j}}{\Delta x_1} + \frac{u_{2_{i,j+\frac{1}{2}}} s^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} - u_{2_{i,j-\frac{1}{2}}} s^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}}{\Delta x_2}.$$

As the slope of Fromm's method $\Delta x_1 \frac{\partial s}{\partial x_1}$ leads to oscillations and therefore instabilities ([66]) we describe in the following section how these slopes substituted in equation (8.24) using the Van Leer's limiter.

Van Leer's limiter

In order to avoid oscillations in equation (8.24) Fromm slopes are replaced by Van Leer slops. As a consequence, the scheme retains the high-order accuracy of Fromm's scheme in smooth regions, but where discontinuities are detected, the discretisised evolution equation drops to first-order accuracy. The Van Leer's limiters are able to detect discontinuities and then modifies the behaviour of the scheme accordingly.

First, one has to detect which sign the slops in the two directions have. As the computation for both directions (x_1 and x_2) are quite similar we show the explicit computation only for the x_1 direction. We computes the sign of the following expression

$$\varphi_{x_1,i,j} = (s_{i+1,j} - s_{i,j})(s_{i,j} - s_{i-1,j})$$

to detect whether to switch on or off the Van Leer slope:

$$\Delta_{x_1}^{VL} s_{i,j} = \begin{cases} S_{x_1,i,j} \min\{\text{slope}_{x_1}\} & \text{if} \quad \varphi_{x_1,i,j} > 0 \\ 0 & \text{if} \quad \varphi_{x_1,i,j} < 0 \end{cases},$$

where we used the short cut

$$\{\text{slope}_{x_1}\} = \{2|s_{i,j} - s_{i-1,j}|, \frac{1}{2}|s_{i+1,j} - s_{i-1,j}|, 2|s_{i,j} - s_{i+1,j}|\}.$$

The variable $S_{x_1,i,j}$ represents the sign of the gradient:

$$S_{x_1,i,j} = sign(s_{i+1,j} - s_{i-1,j}).$$

Using the analogous computation the slope $\Delta x_1 \frac{\partial s}{\partial x_1}$ in equation (8.24) is reformulated into

$$\begin{aligned} s_{i+\frac{1}{2},j}^{n+\frac{1}{2}} &= s_{i,j}^n + \frac{1}{2}\left(1 - u_{i+\frac{1}{2},j}\frac{\Delta t}{\Delta x_1}\right)\Delta_{x_1}^{VL} s_{i,j} - \frac{1}{2}\Delta t\ s_{i,j}^n \left(\frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x_1}\right) \\ & -\frac{1}{2}\Delta t\left(\frac{u_{2_{i,j+\frac{1}{2}}} s_{i,j+\frac{1}{2}}^{upwind} - u_{2_{i,j-\frac{1}{2}}} s_{i,j-\frac{1}{2}}^{upwind}}{\Delta x_2}\right), \end{aligned} \tag{8.25}$$

where s^{upwind} denotes the upwind value of s that is defined in the following and depicted in figure 8.6. It is a term that, depending on the direction of the velocity u, selects the appropriate neighbour s for the computation in such a way that the CFL-condition [1] holds. The computation of the two upwind terms is given by:

$$s_{i,j+\frac{1}{2}}^{upwind} = \begin{cases} s_{i,j}^n & if \quad u_{2_{i,j+\frac{1}{2}}} > 0 \\ s_{i,j+1}^n & if \quad u_{2_{i,j+\frac{1}{2}}} < 0 \end{cases}$$

and

$$s_{i,j-\frac{1}{2}}^{upwind} = \begin{cases} s_{i,j-1}^n & if \quad u_{2_{i,j-\frac{1}{2}}} > 0 \\ s_{i,j}^n & if \quad u_{2_{i,j-\frac{1}{2}}} < 0 \end{cases}.$$

Corrector Step

The *corrector step* within this scheme is implemented by the conservative differencing for the new value $s_{i,j}^{n+1}$ at the time $(n+1)\Delta t$. It is computed with equation (8.22) using the previously calculated half-step values $s_{i-\frac{1}{2},j}$, $s_{i+\frac{1}{2},j}$, $s_{i,j-\frac{1}{2}}$ and $s_{i,j+\frac{1}{2}}$ of the *predictor step.*

The above presented predictor-corrector scheme is used in the following to discretise space-time dependent evolution equations.

---

[1] The Fromm scheme [50] is a second-order scheme and is stable for $0 \leq \sigma_{CFL} \leq 1$.

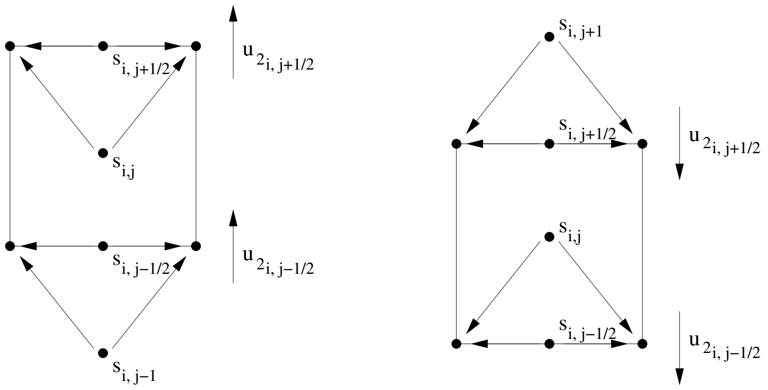


Figure 8.6: Upwind. **Left:** Upwind scheme for the positive velocity u_2 **Right:** Upwind scheme for the negative velocity u_2

8.2.3 Discretised Vorticity Transport Equation

In this section we discuss our implementation of the incompressible vorticity transport equation, which we used in [118, 117]. The vorticity transport equation is given by

$$\frac{\partial \omega}{\partial t} + (u \cdot \nabla)\omega = \nu \Delta \omega, \tag{8.26}$$

where $u = (u_1, u_2)^\top$, $u = u(x,t)$, $x = (x_1, x_2)^\top$ and ν represents the kinematic viscosity (diffusion) coefficient. For our purposes, we think of the vorticity $\omega = \omega(x,t)$ as a scalar quantity which is transported or advected by an incompressible velocity field u. The velocity field is divergence-free so that

$$\nabla \cdot u = 0.$$

The vorticity transport equation represents a parabolic equation and consist of two parts. The hyperbolic advection part

$$\frac{\partial \omega}{\partial t} + (u \cdot \nabla)\omega = 0$$

and the elliptic diffusion part

$$\frac{\partial \omega}{\partial t} = \nu \Delta \omega.$$

As we know how to handle the advection and diffusion part separately we have to combine these skills to a hybrid approach which solves the full equation. This results in a second-order accurate explicit-implicit method. The explicit second-order Fromm-Van-Leer scheme is used for the advection part of the vorticity transport equation and the implicit Crank-Nicholson second-order temporal differencing scheme for the diffusion part of the equation.

The main task we pursue in the following is to combine the explicit Fromm-Van Leer scheme (cf. sec. 8.2.2) with the implicit unconditionally stable Crank-

Nicholson scheme (cf. sec. 8.1.6) to achieve spatial and temporal stability.

Crank-Nicholson Scheme

In order to obtain an explicit formulation for the advection part we try to define the advection part independently of time step $t = n + 1$ of the implicit Crank Nicholson. Therefore we introduce a half-time step $t = n + \frac{1}{2}$. For the diffusion part, on the contrary, we use an implicit scheme - therefore, we are allowed to define ω at the time step $t = n + 1$.

$$\begin{aligned}\frac{\omega_{i,j}^{n+1} - \omega_{i,j}^{n}}{\Delta t} &= -(u \cdot \nabla)\omega_{i,j}^{n+\frac{1}{2}} + \nu \Delta^h (\omega^n + \omega^{n+1})_{i,j} \\ \omega_{i,j}^{n+1} &= \omega_{i,j}^{n} - \Delta t (u \cdot \nabla)\omega_{i,j}^{n+\frac{1}{2}} + \Delta t \nu \Delta^h (\omega^n + \omega^{n+1})_{i,j} \\ \left(I - \frac{\Delta t}{2}\nu \Delta^h\right)\omega_{i,j}^{n+1} &= \omega_{i,j}^{n} - \Delta t (u \cdot \nabla)\omega_{i,j}^{n+\frac{1}{2}} + \frac{\Delta t}{2}\nu \Delta^h \omega_{i,j}^{n}, \qquad (8.27)\end{aligned}$$

where $\Delta^h \omega_{ij} = \frac{\omega_{i+1,j} + \omega_{i-1,j} - 4\omega_{i,j} + \omega_{i,j+1} + \omega_{i,j-1}}{h^2}$ represents the discretised form of the standard five-point Laplacian.

Note that equation (8.27) is written in operator form where a linear operator L acts on $\tilde{\omega}$ and is equal to b which is defined by the right-hand side of the equation:

$$L\tilde{\omega} = b.$$

The linear operator is

$$L = \left(I - \frac{\Delta t}{2}\nu \Delta^h\right).$$

Using the standard five-point Laplacian we write $L\tilde{\omega}$ at any point (i, j) as

$$(L\tilde{\omega})_{i,j} = \left(1 + \frac{4\Delta t \nu}{2h^2}\right)\tilde{\omega}_{i,j} - \frac{\Delta t \nu}{2h^2}\left(\tilde{\omega}_{i+1,j} + \tilde{\omega}_{i-1,j} + \tilde{\omega}_{i,j+1} + \tilde{\omega}_{i,j-1}\right).$$

The right-hand-side of equation (8.27) is therefore

$$b = \omega_{i,j}^{n} - \Delta t (u \cdot \nabla)\omega_{i,j}^{n+\frac{1}{2}} + \frac{\Delta t}{2}\nu \Delta^h \omega_{i,j}^{n}. \qquad (8.28)$$

In the following we are mainly interested in the discretisation of the explicit part $(u \cdot \nabla)\omega_{i,j}^{n+\frac{1}{2}}$ of equation (8.28) to the time step $n + \frac{1}{2}$. Therefore we use the second-order Fromm Van Leer scheme as follows:

Fromm Van Leer Scheme

When one tries to discretise the equation (8.28), the main difficulties occur within the discretisation of the term $\nabla\ (u\ \omega)_{i,j}^{n+\frac{1}{2}}$. Note that for the discrete formulation, we assume that the scalar ω is located at the cell centre and that velocities u enclosing the cell are defined on the cell edges (compare fig. 8.7).

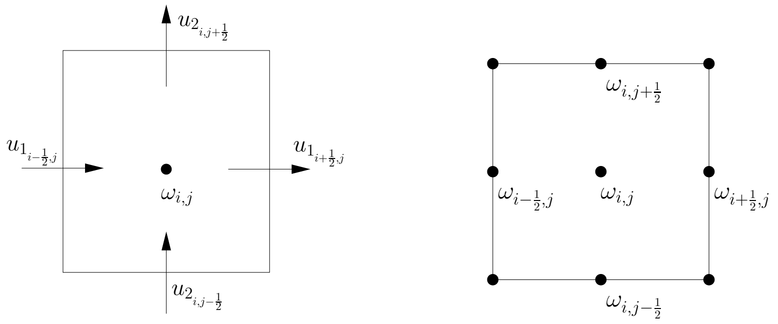


Figure 8.7: Finite differences using Fromm scheme. **Left:** Velocity components u_1 and u_2 on the cell edges. **Right:** Vorticity ω on the cell edges.

As we are considering the incompressible vorticity transport equation, the following identity holds true:

$$(u \cdot \nabla)\omega = \nabla \cdot (u\omega).$$

The standard finite-difference approximation for this term is given by

$$\nabla\ (u\ \omega)_{i,j}^{n+\frac{1}{2}} = \frac{u_{1_{i+\frac{1}{2},j}}\omega_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - u_{1_{i-\frac{1}{2},j}}\omega_{i-\frac{1}{2},j}^{n+\frac{1}{2}}}{\Delta x_1} + \frac{u_{2_{i,j+\frac{1}{2}}}\omega_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - u_{2_{i,j-\frac{1}{2}}}\omega_{i,j-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x_2},$$

(cf. [36]). For this the computation of the vorticities at the four cell boundaries is needed. We will only derive the formula for the approximation of the vorticity $\omega_{i+\frac{1}{2},j}^{n+\frac{1}{2}}$ at the right edge. But note that the derivation of the other vorticities is analogous. A Taylor expansion for the right edge leads to the following approximation:

$$\omega_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \approx \omega_{i,j}^{n} + \frac{1}{2}\Delta x_1 \frac{\partial \omega}{\partial x_1} + \frac{1}{2}\Delta t \frac{\partial \omega}{\partial t}. \tag{8.29}$$

Using the vorticity transport equation (8.26) within this approximation one gets:

$$\begin{aligned}
\omega_{i+\frac{1}{2},j}^{n+\frac{1}{2}} &= \omega_{i,j}^{n} + \frac{1}{2}\Delta x_1 \frac{\partial \omega}{\partial x_1} + \frac{1}{2}\Delta t\left(-u_1\frac{\partial \omega}{\partial x_1} - \omega\frac{\partial u_1}{\partial x_1} - \frac{\partial (u_2\omega)}{\partial x_2} + \frac{1}{2}\Delta t \nu \Delta\omega\right)\\
&= \omega_{i,j}^{n} + \frac{1}{2}\left(1 - u_1\frac{\Delta t}{\Delta x_1}\right)\Delta x_1 \frac{\partial \omega}{\partial x_1} - \frac{1}{2}\Delta t\ \omega\ \frac{\partial u_1}{\partial x_1}\\
&\quad -\frac{1}{2}\Delta t \frac{\partial (u_2\omega)}{\partial x_2} + \frac{1}{2}\Delta t \nu \Delta\omega.
\end{aligned}$$

The discretisation of this equation results in:

$$\begin{aligned}\omega^{n+\frac{1}{2}}_{i+\frac{1}{2},j} &= \omega^{n}_{i,j} + \frac{1}{2}\left(1 - u1_{i+\frac{1}{2},j}\frac{\Delta t}{\Delta x_1}\right)\Delta^{VL}_{x_1}\omega_{i,j} - \frac{1}{2}\Delta t\, \omega^{n}_{i,j}\left(\frac{u1_{i+\frac{1}{2},j} - u1_{i-\frac{1}{2},j}}{\Delta x_1}\right)\\ &-\frac{1}{2}\Delta t\left(\frac{u2_{i,j+\frac{1}{2}}\omega^{upwind}_{i,j+\frac{1}{2}} - u2_{i,j-\frac{1}{2}}\omega^{upwind}_{i,j-\frac{1}{2}}}{\Delta x_2}\right) - \frac{1}{2}\Delta t\nu(\Delta\omega^{n})_{i,j},\end{aligned}$$

where $\Delta^{VL}_{x_1}\omega_{i,j}$ represents the Van Leer slope (see also sec. 8.2.1 and 8.2.2). The upwind value of ω is denoted by

$$\omega^{upwind}_{i,j+\frac{1}{2}} = \begin{cases} \omega^{n}_{i,j} & if \quad u2_{i,j+\frac{1}{2}} > 0 \\ \omega^{n}_{i,j+1} & if \quad u2_{i,j+\frac{1}{2}} < 0. \end{cases}$$

Again, the upwind variable selects – depending on the direction of the velocity u – the appropriate neighbour ω in the computation such that the CFL-condition holds in the numerical scheme.

8.3 Numerical Methods for Non-Linear Conservation Laws

8.3.1 Godunov Schemes

Godunov [54] extended 1959 the first-order conservative upwind scheme of Courant, Isaacson and Rees [38] for linear systems to a scheme for non-linear systems of hyperbolic conservation laws

$$\frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = 0. \tag{8.30}$$

The solution in Godunov's Method is assumed to be a piecewise constant function over each mesh cell at a fixed time $t = n\Delta t$. At a particular time level a pair of constant states (u^n_i, u^n_{i+1}) is considered, which are separated by a discontinuity at the cell boundary or interface $x_{i+\frac{1}{2}}$. The flow evolution in each cell for the next time step $t = (n+1)\Delta t$ results from the interactions between the cells at the cell interfaces. At the cell interface two different fluid states u_l for the left side and u_r for the right side are defined. A Riemann problem [2] can

---

[2] The Riemann problem for a $m \times m$ non-linear hyperbolic system (8.30) is an initial-value problem with the initial data given in the following special form

$$u(x,0) = \begin{cases} u_l & x < 0 \\ u_r & x > 0. \end{cases}$$

Here u_l and u_r represent two real initial values for the left and the right part of the initial state, respectively. For this problem the solution is known to consists of $m+1$ constant states, which are separated by m waves. These waves can be considered as isolines of the function be stated which defines the interaction between the left and right side of the cell interface. The solution of this problem leads to the values u_l and u_r. That means that the initial conditions at the time $t = n\Delta t$ at the cell boundaries corresponds to the exact solution of the respectively local Riemann problem. The solution of the Riemann problem resolved the local interaction between u_l and u_r.

Numerically, each wave propagates information – complying the upwind scheme – over the time interval Δt. The solution of the Riemann problem provides values at the contact discontinuities formed by shock waves or expansion fans. Note that this could result in different values for a single cell at its cell boundaries, violating the constraint of a piecewise constant state in the cell. Furthermore, to confine this interaction only between adjacent cells the time interval must be limited by the CFL-conditon thereby preventing the interference with adjacent Riemann problems.

To obtain the new piecewise constant approximation at the new time interval $t = (n + 1)\Delta t$ the fluid states obtained at the all boundaries for each cell are just averaged. By the averaging details of the exact Riemann solutions get lost. Therefore, one do not need exact solutions of the Riemann problems that are computationally expensive to obtain. As a consequence we use approximate Riemann solutions at the cell interfaces.

In the following we describe the general, explicit first-order conservative Godunov method in order to lay the foundations for the second-order Godunov method which we used in our numerical implementation of the Burgers equation (3.1), (4.4), and (6.2).

The General Godunov Method

We want to solve the general initial value problem

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial u} &= 0 \\ u(x,0) &= u_0, \end{aligned} \tag{8.31}$$

with respect to the conserved quantities u and fluxes $F(u)$ using the first-order Godunov's method [43]. The conserved quantities u are evaluated in each cell of size Δx at each time step t^n. To compute the solution of (8.31) at the next time step t^{n+1} three main steps are relevant:

First, one defines a piecewise linear interpolation function $\hat{u}$, such that

---

which also describe the discontinuities which build a shock. They are known as contact waves, or transition waves such as rarefaction waves.

$$u_i^n \Delta x = \int\limits_{(i-\frac{1}{2})\Delta x}^{(i+\frac{1}{2})\Delta x} \hat{u}(x, t^n) dx.$$

The piecewise approximation is then an average of the solution over each appropriate cell $(i - \frac{1}{2}, i + \frac{1}{2})$ and the integral form of the conservation law (8.31) is used to derive an explicit conservative formula

$$u_i^{n+1} = u_i^n + \frac{\Delta t}{\Delta x}\left(F_{i-\frac{1}{2}} - F_{i+\frac{1}{2}}\right), \tag{8.32}$$

as described in section 8.2.

In the second step, the numerical flux over the cell boundaries

$$F_{i+\frac{1}{2}} = F(u^{RP}_{i+\frac{1}{2}}),$$

where $u^{RP}_{i+\frac{1}{2}}$ denotes the solution of the Riemann problem at the cell interface, are obtained by solving the appropriate Riemann problem

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} &= 0 \\ u(x, 0) &= \left\{ \begin{array}{lll} u_l & \text{if} & x < 0 \\ u_r & \text{if} & x > 0 \end{array} \right. . \end{aligned} \tag{8.33}$$

The Riemann problem (8.33) is evaluated at $\frac{x}{t} = 0$ at each cell interface $x_{i+\frac{1}{2}}$. That means that the flux functions F incorporate the solutions of the Riemann problem.

The third step consists of averaging the state variables at the new time

$$u_i^{n+1} = \frac{1}{\Delta x} \int\limits_{(i-\frac{1}{2})\Delta x}^{(i+\frac{1}{2})\Delta x} \hat{u}(x, t^{n+1}) dx.$$

These values define the piecewise constant solution at the time $t + 1$ and the computation process is repeated to obtain the solution for the next time step. In the following we are concerned with the question how to choose the length of the time step.

The choice of the time step Δt: We applied the conservative formula (8.32) to compute the intercell fluxes. The spacial mesh length Δx is given by the image grid size, usually $\Delta x = 1$. The choice of the time step length Δt in the conservative formula depends on the stability condition of the particular scheme. The choice of Δt in the Godunov method depends on the restriction defined by the CFL number σ_{CFL}. The choice for non-linear schemes is difficult as for each time level multiple wave speeds exist and therefore multiple CFL numbers. To overcome this problem we compute

$$a_{max} = \frac{\partial F}{\partial u}\Big|_{max}$$

the maximum wave speed within the region of interest at time level n and the maximum CFL number is then

$$\sigma_{CFL_{max}} = a_{max}\frac{\Delta t}{\Delta x}, \tag{8.34}$$

with

$$0 < \sigma_{CFL_{max}} \leq 1. \tag{8.35}$$

A reasonable choice of the time step Δt, holding the CFL condition, is

$$\Delta t = \frac{\sigma_{CFL_{max}}}{a_{max}}\Delta x. \tag{8.36}$$

That means that no wave present in the region of interest at time n travels more than a distance Δx within the time Δt. The information transport within one time step is not greater than Δx. Note that the closer the CFL number $\sigma_{CFL_{max}}$ is to 1, the more efficient is the computation.

8.3.2 Second-Order Unsplit Godunov Method

Colella and Puckett [36] developed a so called unsplit Godunov method for general systems of conservation laws using a predictor-corrector approach of second-order to obtain higher-order accuracy of the numerical solution. The higher-order approximation is based on a second-order approximation to the solution at the cell boundaries at time $t^{n+\frac{1}{2}}$ of the left and right states. These states are used to compute the numerical flux F at the cell interfaces. The resulting conservative finite difference scheme (8.50) is then a second-order approximation.

In the following we are concerned with the initial value problem

$$\begin{aligned} \frac{\partial u}{\partial t} + \nabla \cdot F(u) &= 0 \\ u(x,0) &= u_0, \end{aligned} \tag{8.37}$$

where $x = (x_1, x_2)$, $u = u(x,t)$, and $u = (u_1, u_2)$ a function $u \in \mathbb{R}^2 \times [0,T]$ and F the flux function. This system of equations is written as a single vector equation

$$\frac{\partial u}{\partial t} + \frac{\partial F_1(u)}{\partial x_1} + \frac{\partial F_2(u)}{\partial x_2} = 0. \tag{8.38}$$

We assume, that we know or can compute the cell average value $u_{i,j}^n$ of the solution over each cell $(i\Delta x_1, j\Delta x_2)$

$$u_{i,j}^n = \frac{1}{\Delta x_1 \Delta x_2} \int\limits_{(i-\frac{1}{2},j)\Delta x_1}^{(i+\frac{1}{2},j)\Delta x_1} \int\limits_{(i,j-\frac{1}{2})\Delta x_2}^{(i,j+\frac{1}{2})\Delta x_2} u(x_1, x_2, t^n)\, dx_1 dx_2. \tag{8.39}$$

Below we describe Godunov's second-order predictor-corrector method. It allows to compute the solutions $u_{i,j}^{n+1}$ at the time $t^{n+1} = (n+1)\Delta t$.

1. Computation of the second-order approximations
   a) Construction of a linear monotonised approximation
   b) Predictor step: Extrapolation to the cell interfaces

2. Solution of the Riemann problem at the cell interfaces

3. Corrector step: Using conservative finite differencing to update the solution

These steps are described in more detail below:

Linear Monotonised Approximation

Harten [61] expanded the central difference approximation of the spatial derivative resulting in the following approximation:

$$\Delta^{x_1} u_{i,j} \approx \Delta x_1 \frac{\partial u}{\partial x_1}\Big|_{(i\Delta x_1, j\Delta x_2)} = \sum_k \alpha_{k_j} r_k^{x_1}. \tag{8.40}$$

For this the author used the right eigenvectors of the Jacobian matrix $\frac{\partial F}{\partial u}$ [3] of the linearised equation (8.41). Then the cell amplitude α_{k_j} is computed as follows

$$\alpha_{k_j} = \begin{cases} sign(\alpha_{k_j}^C) \min\{2|\alpha_{k_j}^L|, |\alpha_{k_j}^C|, 2|\alpha_{k_j}^R|\} & \text{if } \left(\alpha_{k_j}^L \alpha_{k_j}^R > 0\right) \\ 0 & \text{if } \left(\alpha_{k_j}^L \alpha_{k_j}^R \le 0\right), \end{cases} \tag{8.42}$$

where the amplitudes are defined by

$$\begin{aligned} \alpha_{k_j}^L l_k^{x_1} &= \frac{1}{2}(u_{i,j} - u_{i-1,j}), \\ \alpha_{k_j}^C l_k^{x_1} &= \frac{1}{2}(u_{i+1,j} - u_{i-1,j}), \\ \alpha_{k_j}^R l_k^{x_1} &= \frac{1}{2}(u_{i+1,j} - u_{i,j}), \end{aligned}$$

---

[3] By applying the chain rule to the second term of (8.30) we rewrite this equation in the following (quasi-linearised) form:

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial u}\frac{\partial u}{\partial x} = 0. \tag{8.41}$$

The term $\frac{\partial F}{\partial u}$ denotes the Jacobian matrix of the flux function $F(u)$.

and $l_k^{x_1}$ denotes the left eigenvectors of the Jacobian matrix $\frac{\partial F}{\partial u}$. This technique is related to Van Leer's limiter construction described in section 8.2.1.

These computations serve as an approximation for a spatial derivative that is used within the following step.

Extrapolation to the cell interfaces

In this section we are concerned with the predictor step. That means that the occurring fluxes are extrapolated to the cell boundaries. A Riemann problem for each edge of a cell has to be solved for the calculation of the edge fluxes. We demonstrate only the computation for the left state. The right, top and down states are computed analogous. Using a Taylor expansion one obtains

$$\begin{aligned} u_{i+\frac{1}{2},j}^{n+\frac{1}{2}}\Big|_L &= u_{i,j}^n + \frac{\Delta x_1}{2}\frac{\partial u}{\partial x_1}\Big|_{(i,j)}^n + \frac{\Delta t}{2}\frac{\partial u}{\partial t}\Big|_{(i,j)}^n \\ &= u_{i,j}^n + \frac{\Delta x_1}{2}\frac{\partial u}{\partial x_1}\Big|_{(i,j)}^n - \frac{\Delta t}{2}\Big(\frac{\partial F_1(u)}{\partial x_1} + \frac{\partial F_2(u)}{\partial x_2}\Big)\Big|_{(i,j)}^n \\ &= u_{i,j}^n + \frac{1}{2}\Big(\mathbb{1} - \frac{\partial F_1(u)}{\partial u}\frac{\Delta t}{\Delta x_1}\Big)\Delta x_1 \frac{\partial u}{\partial x_1}\Big|_{(i,j)}^n - \frac{\Delta t}{2}\frac{\partial F_2(u)}{\partial x_2}\Big|_{(i,j)}^n . \end{aligned} \tag{8.43}$$

Here the term $\Delta x_1 \frac{\partial u}{\partial x_1}\Big|_{(i,j)}^n$ is approximated by the monotonised differences (8.40). It is known that for non-linear problems, components which corresponds to characteristics which do not propagate towards the zone edges must vanish. For the upwind computation Colella and Puckett [36] defined the following characteristic projection operators

$$P_L(S) = \sum_{\substack{k \\ \lambda_k^{x_1}(u_{i,j}) > 0}} \Big(l_k^{x_1}(u_{i,j}) \cdot S\Big) r_k^{x_1}(u_{i,j})$$

$$P_R(S) = \sum_{\substack{k \\ \lambda_k^{x_1}(u_{i,j}) < 0}} \Big(l_k^{x_1}(u_{i,j}) \cdot S\Big) r_k^{x_1}(u_{i,j}), \tag{8.44}$$

where $\lambda_k^{x_1}(u_{i,j})$ represents the k-th real eigenvalue of the Jacobian matrix $\frac{\partial F}{\partial u}$. The last term $\frac{\Delta t}{2}\frac{\partial F_2(u)}{\partial x_2}\Big|_{(i,j)}^n$ that occurs in the equation (8.43) is known in the literature as transverse derivative. Here it is approximated by the differences of Godunov fluxes in an upwind manner as also defined in (8.2.2). Finally the expression for the computation of the right and left states becomes

$$\begin{aligned} u^{n+\frac{1}{2}}_{i+\frac{1}{2},j}\Big|_L &= u^n_{i,j} + \frac{1}{2}P_L\Big(\mathbb{1} - \frac{\partial F_1(u)}{\partial u}\frac{\Delta t}{\Delta x_1}\Big)\Delta^{x_1}u_{i,j}\Big|^n_{(i,j)} \\ &\quad -\frac{\Delta t}{2}\Big(F_2(\hat{u}_{i,j+\frac{1}{2},L} - F_2(\hat{u}_{i,j-\frac{1}{2},L}\Big)\Big|^n_{(i,j)} \end{aligned} \tag{8.45}$$

$$\begin{aligned} u^{n+\frac{1}{2}}_{i+\frac{1}{2},j}\Big|_R &= u^n_{i+1,j} - \frac{1}{2}P_R\Big(\mathbb{1} + \frac{\partial F_1(u)}{\partial u}\frac{\Delta t}{\Delta x_1}\Big)\Delta^{x_1}u_{i+1,j}\Big|^n_{(i+1,j)} \\ &\quad -\frac{\Delta t}{2}\Big(F_2(\hat{u}_{i,j+\frac{1}{2},R} - F_2(\hat{u}_{i,j-\frac{1}{2},R}\Big)\Big|^n_{(i+1,j)}, \end{aligned} \tag{8.46}$$

where the fluxes $F_2(\hat{u}_{i,j-\frac{1}{2},L})$ and $F_2(\hat{u}_{i,j-\frac{1}{2},R})$ are computed using the conservation formula (8.51) and the projection operator (8.44) as in (8.43).

Solution for the Riemann Problem

In this part we are concerned with the computation of the quantities u at the cell boundaries. Note that these are used to compute the fluxes. In the previous step the left and right states $u^{n+\frac{1}{2}}_{i+\frac{1}{2},j}\Big|_L$ and $u^{n+\frac{1}{2}}_{i+\frac{1}{2},j}\Big|_R$ were computed. These are used as initial data for the left and right states within the Riemann Problem

$$u(x,0) = \left\{ \begin{array}{ll} u_L & \text{if} \quad x < 0 \\ u_R & \text{if} \quad x > 0 \end{array} \right. . \tag{8.47}$$

A solution of the Riemann problem has to satisfy the following conditions:

i) The solution $u(x,t)$ of the Riemann problem is a function of the similarity variable $\frac{x}{t}$ only and is known as similarity solution $u(\frac{x}{t})$ (c.f. (8.33)).

ii) The similarity solution $u(\frac{x}{t})$ has to satisfy the entropy condition

$$\lambda(u_l) > S = \frac{F(u_r) - F(u_l)}{(u_r - u_l)} > \lambda(u_r),$$

where λ represents the eigenvalues of the Jacobian matrix $\frac{\partial F}{\partial u}$ and S the wave speed.

iii) Each wave family k in a characteristic field λ_k is either linearly degenerate

$$\nabla\lambda(u) \cdot \nu(u) = 0 \tag{8.48}$$

or genuinely non-linear

$$\nabla\lambda(u) \cdot \nu(u) \neq 0, \tag{8.49}$$

where $\nu(u)$ is the corresponding eigenvector to $\lambda(u)$.

We consider the k-th wave and define the following solutions:

$$u(x,t) = \begin{cases} u_L & \text{if} \quad \frac{x}{t} < \lambda_{k_L}, k = 1, ..., m \\ u_{k-1} + \frac{\frac{x}{t} - \lambda_{k_L}}{\lambda_{k_R} - \lambda_{k_L}}(u_{k-1} - u_k) & \text{if} \quad \lambda_{k_L} \leq \frac{x}{t} < \lambda_{k_R}, k = 1, ..., m \\ u_R & \text{if} \quad \frac{x}{t} > \lambda_{k_R}, k = 1, ..., m \end{cases},$$

where for each wave the left and right states of the k-th eigenvalue λ_k, $k = 1, ..., m$ has to be determined. If the k-th wave is linear degenerated (c.f. (8.48)) then

$$\lambda_{k_L} = \lambda_{k_R} = \frac{1}{2}(\lambda_k(u_{k-1}) + \lambda_k(u_k)).$$

If the k-th wave is genuinely non-linear (c.f. (8.49)) and $\lambda_k(u_{k-1}) < \lambda_k(u_k)$ then

$$\begin{aligned} \lambda_{k_L} &= \lambda_k(u_{k-1}) \\ \lambda_{k_R} &= \lambda_k(u_k). \end{aligned}$$

However, a full discussion of shocks and shock formations is beyond the scope of this thesis and therefore, we refer to the books of Chorin and Marsden [32], Lax and Wendroff [86], LeVeque [87], Toro [134] and Thomas [130] for further studies of this topic.

Conservative Finite Differencing

The conservative formula is discretised by the following expression

$$u_{i,j}^{n+1} = u_{i,j}^n - \frac{\Delta t}{\Delta x_1}\Big(F_1(\hat{u}_{i+\frac{1}{2},j}^n) - F_1(\hat{u}_{i-\frac{1}{2},j}^n)\Big) - \frac{\Delta t}{\Delta x_2}\Big(F_2(\hat{u}_{i,j+\frac{1}{2}}^n) - F_2(\hat{u}_{i,j-\frac{1}{2}}^n)\Big), \tag{8.50}$$

where $F(\hat{u}_{i+\frac{1}{2},j}^n)$ represents the flux function which is determined by

$$F_1(\hat{u}_{i+\frac{1}{2},j}^n) = \lambda_{i,j}^{x_1}\hat{u}_{i+\frac{1}{2},j}^n.$$

Note that this equation depends on $\hat{u}_{i+\frac{1}{2},j}^n$ which is the Riemann solution and can be calculated using the approximation with the knowledge of the previous section. To guaranty the stability of this computation scheme one can specify the CFL-condition (c.f. sec. 8.1.3) in the following form:

$$\max_{(i,j)}\left(\left|\lambda_{i,j}^{x_1}\frac{\Delta t}{\Delta x_1}\right|, \left|\lambda_{i,j}^{x_2}\frac{\Delta t}{\Delta x_2}\right|\right) \leq 1.$$

8.3.3 Discretisation of the Adjoint Equation

In this section we are concerned with the implementation of the time-dependent adjoint system (8.51) (cf. (4.7*b*) of chapter 4) on the domain $\Omega \times [0, T]$ by using a second-order predictor-corrector finite difference scheme.

$$\left\{ \begin{array}{lcl} \frac{\partial w_1}{\partial t} + u_1 \frac{\partial w_1}{\partial x_1} + u_2 \frac{\partial w_1}{\partial x_2} + w_1 \frac{\partial u_2}{\partial x_2} - w_2 \frac{\partial u_2}{\partial x_1} & = & D_1(I, u) \\ \frac{\partial w_2}{\partial t} + u_1 \frac{\partial w_2}{\partial x_1} + u_2 \frac{\partial w_2}{\partial x_2} + w_2 \frac{\partial u_1}{\partial x_1} - w_1 \frac{\partial u_1}{\partial x_2} & = & D_2(I, u) \end{array} \right. \tag{8.51}$$

The data term $D = (D_1, D_2)^\top$ is independent from the adjoint variable and is in our case defined as

$$D(I, u) := (\nabla I^\top u + I_t) \nabla I - \alpha \Delta u, \tag{8.52}$$

and is precomputed for the entire image sequence.

As before the discretisation domain $\Omega \times [0, T]$ is divided into cells like depicted in figure 8.1 in section 8.1. The adjoint quantities $w = (w_1, w_2)^\top$ are defined at the centres of the individual cells: For example $w^n_{1_{i,j}}$ represents the adjoint variable at the position $(x_1, x_2) = (i, j)$ within the grid at time $t = n$. The quantities at the cell boundaries which lie between the cells (i, j) and $(i, j+1)$ are denoted by $w^n_{1_{i+\frac{1}{2},j}}$ as depicted in figure 8.8.

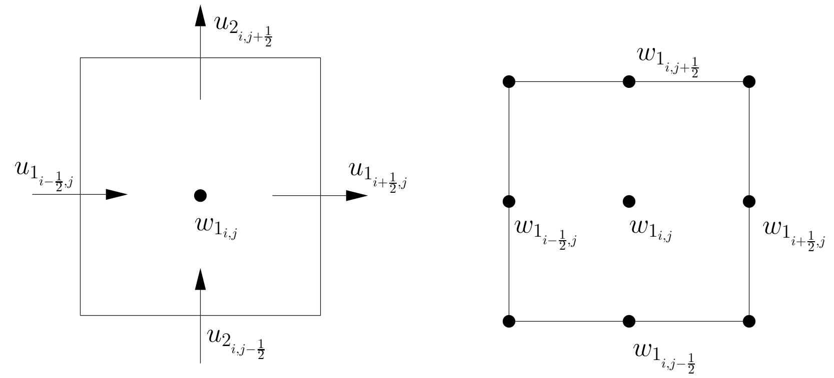


Figure 8.8: One cell of the domain Ω with respect to the velocity components and the adjoint quantities. **Left:** Velocity components u_1 and u_2 on the cell edges. **Right:** Notation of the adjoint quantities of the cell. Quantities on the boundary are subscripted by half indices.

The conservation form[4] for this problem can be written as follows:

$$\begin{aligned} w^{n+1}_{1_{i,j}} &= w^n_{1_{i,j}} + \frac{\Delta t}{\Delta x}(F_{in} - F_{out}) \\ &= w^n_{1_{i,j}} + \frac{u_1 \Delta t}{\Delta x_1}(w^{n+\frac{1}{2}}_{1_{i-\frac{1}{2},j}} - w^{n+\frac{1}{2}}_{1_{i+\frac{1}{2},j}}) + \frac{u_2 \Delta t}{\Delta x_2}(w^{n+\frac{1}{2}}_{1_{i,j-\frac{1}{2}}} - w^{n+\frac{1}{2}}_{1_{i,j+\frac{1}{2}}}). \end{aligned}$$

---

[4] As the quantity w is physically conserved over time.

The numerical fluxes F_{in} and F_{out} are obtained by solving the predictor step, which consists of the computation of the adjoint quantities at the cell edges (cf. fig. 8.8).

Note that all fluxes are evaluated and differenced at the same time. We employ the upwind finite difference method – also known as Fromm Van Leer scheme – that is described in section 8.2.2.

The quantities $w_{1_{i-\frac{1}{2},j}}^{n+\frac{1}{2}}$, $w_{1_{i+\frac{1}{2},j}}^{n+\frac{1}{2}}$, $w_{1_{i,j-\frac{1}{2}}}^{n+\frac{1}{2}}$ and $w_{1_{i,j+\frac{1}{2}}}^{n+\frac{1}{2}}$ at the cell boundaries are approximated using the Taylor series expansion. We will derive this estimation only for one component of the adjoint quantity, namely $w_{1_{i+\frac{1}{2},j}}^{n+\frac{1}{2}}$ which is located at the right edge $(i+\frac{1}{2},j)$. The derivation of the other seven quantities (three for w_1 and four for w_2) is analogous. The linearised Taylor series for this edge yields the following approximation:

$$w_{1_{i+\frac{1}{2},j}}^{n+\frac{1}{2}} \approx w_{1_{i,j}}^{n} + \frac{1}{2}\Delta x_1 \frac{\partial w_1}{\partial x_1} + \frac{1}{2}\Delta t \frac{\partial w_1}{\partial t}. \tag{8.53}$$

In this equation we substitute $\frac{\partial w_1}{\partial t}$ by using the first equation of the adjoint system (8.51) and we obtain:

$$\begin{aligned}
w_{1_{i+\frac{1}{2},j}}^{n+\frac{1}{2}} &= w_{1_{i,j}}^{n} + \frac{\Delta x_1}{2}\frac{\partial w_1}{\partial x_1} + \frac{\Delta t}{2}\left(-u_1\frac{\partial w_1}{\partial x_1} - u_2\frac{\partial w_1}{\partial x_2} - w_1\frac{\partial u_2}{\partial x_2} + w_2\frac{\partial u_2}{\partial x_1}\right) \\
&= w_{1_{i,j}}^{n} + \frac{1}{2}\left(1 - u_1\frac{\Delta t}{\Delta x_1}\right)\Delta_{x_1}^{VL} w_{1_{i,j}} - \frac{\Delta t}{2} w_1 \frac{\partial u_2}{\partial x_2} \\
&\quad + \frac{\Delta t}{2} w_2 \frac{\partial u_2}{\partial x_1} - \frac{\Delta t}{2} u_2 \frac{\partial w_1^{upwind}}{\partial x_2}.
\end{aligned} \tag{8.54}$$

Then the final discretisation leads to the following equation:

$$\begin{aligned}
w_{1_{i+\frac{1}{2},j}}^{n+\frac{1}{2}} &= w_{1_{i,j}}^{n} + \frac{1}{2}\left(1 - u_{1_{i+\frac{1}{2},j}}\frac{\Delta t}{\Delta x_1}\right)\Delta_{x_1}^{VL} w_{1_{i,j}} - \frac{\Delta t}{2\Delta x_2} w_{1_{i,j}}^{n}\left(u_{2_{i,j+\frac{1}{2}}} - u_{2_{i,j-\frac{1}{2}}}\right) \\
&\quad + \frac{\Delta t}{2\Delta x_1} w_{2_{i,j}}^{n}\left(u_{2_{i+\frac{1}{2},j}} - u_{2_{i-\frac{1}{2},j}}\right) - \frac{\Delta t}{2\Delta x_2} u_{2_{i,j}}\left(w_{1_{i,j+\frac{1}{2}}}^{upwind} - w_{1_{i,j-\frac{1}{2}}}^{upwind}\right).
\end{aligned}$$

Here $\Delta_{x_1}^{VL}$ is the *Van Leer slope* and $w_{1_{i,j+\frac{1}{2}}}^{upwind}$ the upwind difference, as defined in section 8.2.2. Upwind schemes only use one direction of information transfer. They consider the flow from upstream to downstream locations. This means that in upwind schemes the data from upstream locations – as perceived from a point (i,j) – are used:

$$w_{1_{i,j+\frac{1}{2}}}^{upwind} = \begin{cases} w_{1i,j}^{n} & if \quad u_{2_{i,j+\frac{1}{2}}} > 0 \\ w_{1_{i,j+1}}^{n} & if \quad u_{2_{i,j+\frac{1}{2}}} < 0 \end{cases}. \tag{8.55}$$

The basic idea behind this is that all methods with an accuracy larger than the order one will produce spurious oscillations in the vicinity of large gradients, while being second-order accurate in regions where the solution is smooth [36]. To prevent such oscillations the slopes of Fromm's method are replaced by the slopes of the Van Leers scheme. The Van Leer scheme *detects* discontinuities and modifies its behaviour in such locations accordingly. The implication of this is that this method retains the high-order accuracy of Fromm's scheme in smooth regions, but near discontinuities the discretised evolution equation drops to first-order accuracy. For further details, we refer to [36].

8.4 Discretisation of the Advection-Diffusion Equation

Our goal is to solve the advection-diffusion equation (6.3) in order to obtain the image intensity function $I(x,t)$. The advection-diffusion equation (6.3) represents a parabolic equation. It can divided into a hyperbolic part

$$\frac{\partial I}{\partial t} + (u \cdot \nabla) I = 0,$$

which is the advection equation and an elliptic part

$$\frac{\partial I}{\partial t} = \nu \Delta I,$$

which represents the diffusion equation. We already know how to handle the advection and diffusion equation separately. Therefore what remains is to combine these two terms into a hybrid approach which solves equation (6.3). The hybrid approach we employ makes use of a semi-implicit method. We use the implicit Crank Nicholson scheme (see also sec. 8.1.6) of second-order temporal differencing for the diffusion part and the explicit second-order Fromm Van Leer scheme for the advection part. Note that the numerical implementation of the diffusion part of equation (6.3) is likely to create oscillations for high viscosity values. The Crank Nicholson scheme, however, is known to be unconditionally stable [36] and provides the desired stability properties. Using it, one can largely avoid spatial and temporal discretisation errors.

Below we describe the numerical implementation of both parts, namely the Crank Nicholson scheme and the Fromm Van Leer scheme. We label a particular mesh cell with ij, while the time level is given as superscript n. A grey value is represented by I_{ij}^n and the boundary between the (i,j)th and $(i,j+1)$th cell is again denoted by $(i, j+\frac{1}{2})$. Furthermore, we define $\Delta x_1 = x_{1_{i+\frac{1}{2},j}} - x_{1_{i-\frac{1}{2},j}}$ and assume that the mesh spacing Δx_1, Δx_2 and the time step Δt are constant.

Implicit Part: Crank Nicholson Scheme

The diffusive part is discretised using the Crank Nicholson scheme. We linearise the advective part of the flow equation and extrapolate it to the next half time step. The advection part is discretised using an explicit scheme (avoiding its dependency on the $(n+1)$-time step). The finite difference approximation for the advection-diffusion equation results in:

$$\begin{aligned} \frac{I_{i,j}^{n+1} - I_{i,j}^{n}}{\Delta t} &= -(u \cdot \nabla) I_{i,j}^{n+\frac{1}{2}} + \nu \frac{1}{2} \Delta (I^n + I^{n+1})_{i,j} \\ I_{i,j}^{n+1} &= I_{i,j}^{n} - \Delta t (u \cdot \nabla) I_{i,j}^{n+\frac{1}{2}} + \frac{\Delta t}{2} \nu \Delta (I^n + I^{n+1})_{i,j} \\ \left(I - \frac{\Delta t}{2} \nu \Delta \right) I_{i,j}^{n+1} &= I_{i,j}^{n} - \Delta t (u \cdot \nabla) I_{i,j}^{n+\frac{1}{2}} + \frac{\Delta t}{2} \nu \Delta I_{i,j}^{n}. \end{aligned} \tag{8.56}$$

Here $\Delta I_{ij} = I_{i+1,j} + I_{i-1,j} - 4I_{i,j} + I_{i,j+1} + I_{i,j-1}$ represents the standard form of a discretised Laplacian operator.

Note that equation (8.56) is written as an equation where a linear operator is applied to $\tilde{I}$ which equals a vector b on the right hand side:

$$L\tilde{I} = b. \tag{8.57}$$

The introduced linear operator is

$$L = \left(I - \frac{\Delta t}{2} \nu \Delta \right).$$

Using the standard (five-point) Laplacian operator we obtain the elements of the vector $L\tilde{I}$ at any point (i, j)

$$\left(L\tilde{I} \right)_{i,j} = \left(1 + \frac{4\Delta t \nu}{2} \right) \tilde{I}_{i,j} - \frac{\Delta t \nu}{2} \left(\tilde{I}_{i+1,j} + \tilde{I}_{i-1,j} + \tilde{I}_{i,j+1} + \tilde{I}_{i,j-1} \right).$$

The solution of (8.57) represents the part which is implicit solved in the algorithm.

Explicit Part: Fromm Van Leer Scheme

The right hand side of equation (8.56) represents the vector b introduced in (8.57) and is therefore:

$$b = I_{i,j}^{n} - \Delta t (u \cdot \nabla) I_{i,j}^{n+\frac{1}{2}} + \frac{\Delta t}{2} \nu \Delta I_{i,j}^{n}.$$

The discretisation of the advection term $(u \cdot \nabla) I_{i,j}^{n+\frac{1}{2}}$ at the half time step $(n+\frac{1}{2})$ is realised with the conservative second-order Fromm Van Leer scheme similar to section 8.2.2.

Finally we obtain a hybridised scheme, which replaced the slope of Fromm's method with Van Leer's slope. The Van Leer function is able to detect discontinuities and it modifies its behaviour accordingly. Recall that the implication of this is that the resulting method retains the high-order accuracy of Fromm's scheme in smooth regions, but where discontinuities occur, the discretised evolution equation falls back to first-order accuracy.

The finite difference approximation for the advection term is expressed as

$$u \cdot \nabla\, I_{i,j}^{n+\frac{1}{2}} = u_{1_{i,j}} \frac{I_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - I_{i-\frac{1}{2},j}^{n+\frac{1}{2}}}{\Delta x_1} + u_{2_{i,j}} \frac{I_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - I_{i,j-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x_2}.$$

To derive the formulas for the image quantities $I_{i+\frac{1}{2},j}^{n+\frac{1}{2}}$ at the cell edges (see fig. 8.9) we use a Taylor expansion. The Taylor expansion for the image quantity I at the right edge omitting higher-order terms leads to the following approximation (again, we demonstrate here only the derivation of the discretisation formula for one edge, the others are derived analogously):

$$I_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \quad = \quad I_{i,j}^{n} + \frac{1}{2}\Delta x_1 \frac{\partial I}{\partial x_1} + \frac{1}{2}\Delta t \frac{\partial I}{\partial t}. \tag{8.58}$$

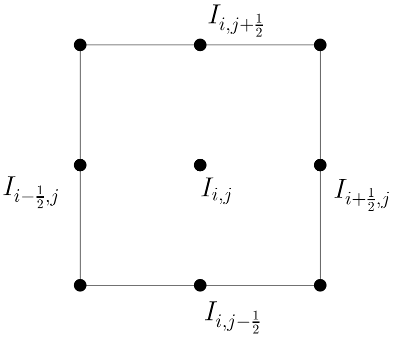


Figure 8.9: One cell of the discretisation domain Ω. Adjoint quantities on the cell edges.

Using equation (6.3) we obtain from (8.58):

$$I_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \quad = \quad I_{i,j}^{n} + \frac{\Delta x_1}{2}\frac{\partial I}{\partial x_1} + \frac{\Delta t}{2}\left(-u_1 \frac{\partial I}{\partial x_1} - u_2 \frac{\partial I}{\partial x_2} + \nu \Delta I\right).$$

Introducing an upwind scheme one finally obtains the following discretised version for the computations of the right edge

$$\begin{aligned} I_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \quad &= \quad I_{i,j}^{n} + \frac{1}{2}\left(1 - u_{1_{i+\frac{1}{2},j}} \frac{\Delta t}{\Delta x_1}\right) \Delta_{x_1}^{VL} I_{i,j} \\ &\quad -\frac{1}{2}\frac{\Delta t}{\Delta x_2} u_{2_{i,j}} \left(I_{i,j+\frac{1}{2}}^{upwind} - I_{i,j-\frac{1}{2}}^{upwind}\right) + \frac{\Delta t}{2}\nu(\Delta I)_{ij}, \end{aligned}$$

where $\Delta_{x_1}^{VL}$ represents the Van Leer slope as described in section 8.2.1 and 8.2.2. The upwind value of I is denoted by I^{upwind}:

$$I^{upwind}_{i,j+\frac{1}{2}} = \begin{cases} I^n_{i,j} & if \quad u_{2_{i,j+\frac{1}{2}}} > 0 \\ I^n_{i,j+1} & if \quad u_{2_{i,j+\frac{1}{2}}} < 0 \end{cases} .$$

As before the upwind scheme means that depending on the direction of the velocity u and therefore of the information transport, the appropriate left or right neighbour cell is used for the computation such that the CFL-condition (sec. 8.1.3) holds. The upwind scheme is explained in more detail in section 8.2.2.

8.5 Finite Element Methods

In this section we will provide a very short overview on finite element methods which represent a suitable discretisation method for the elliptic PDEs that we implemented. Literature to this field: [18, 34, 20].

8.5.1 Variational Formulation for Elliptic Problems

To characterise the problem functions and spaces we define V as a closed subset of a Hilbert space $\mathcal{H}$ and

$$a : \mathcal{H} \times \mathcal{H} \longrightarrow \mathbb{R}$$

as a symmetric bilinear form. Further

$$l : \mathcal{H} \longrightarrow \mathbb{R}$$

defines a linear functional. The space $\mathcal{H}'$ defines the space of continuous and linear functionals of a normalised set $\mathcal{H}$.

To solve the following minimisation problem

$$\inf_{v \in \mathcal{H}} J(v), \tag{8.59}$$

we assume that the energy J can be stated as

$$J(v) = \frac{1}{2} a(v, v) - \langle l, v \rangle. \tag{8.60}$$

To guarantee the solvability of this variational formulation, existence and uniqueness of the solution are essential. The problem (8.60) has one and only one solution if $\mathcal{H}$ is complete, this means closed and restricted, if V is a closed convex subset in $\mathcal{H}$ and if the bilinear form a is symmetric and continuous over the space $\mathcal{H}$ and $\mathcal{H}$-elliptic. This constraints are the basis for the formulation of the Lax-Milgram theorem for convex sets, which guarantees the uniqueness of the solution. Before stating that theorem we need the following definitions:

The bilinear form a is continuous if

$$|a(u,v)| \leq C\|u\| \cdot \|v\|, \quad \forall u,v \in \mathcal{H},$$

with $C > 0$.

The symmetric and continuous bilinear form a is called $\mathcal{H}$-elliptic, if

$$a(v,v) \geq \alpha\|v\|^2, \quad \forall v \in \mathcal{H},$$

with $\alpha > 0$. With that we formulate the

Lax-Milgram Theorem

Let V be a closed and convex subset of a Hilbert space $\mathcal{H}$ and $a : \mathcal{H} \times \mathcal{H} \longrightarrow \mathbb{R}$ be a $\mathcal{H}$-elliptic bilinear form. For every $l \in \mathcal{H}'$, the variational problem

$$J(v) := \frac{1}{2}a(v,v) - \langle l,v\rangle \longrightarrow \min,$$

has an unique solution in $\mathcal{H}$.

To summarise: It is sufficient, that a given symmetric bilinear form is continuous and $\mathcal{H}$-elliptic to obtain an unique solution for the problem (8.59). For the approach of Horn and Schunck (2.8), this was performed in [119].

8.5.2 Ritz-Galerkin Method

For the numerical solution of elliptic problems we use the Ritz-Galerkin method. The problem is formalised as follows: For the energy functional J of a variational formulation one can determine the minimum within some finite-dimensional subspace S_h [112]. The subscript h denotes the discretisation parameter and for $h \longrightarrow 0$ one would expect convergence to the solution of the continuous problem.

The variational problem

$$J(v) := \frac{1}{2}a(v,v) - \langle l,v\rangle \longrightarrow \min_{S_h},$$

has its minimum in $u_h \in S_h$, if the bilinear form

$$a(u_h,v) - \langle l,v\rangle \forall v \in S_h. \tag{8.61}$$

Let $\{\phi_1, \phi_2, ..., \phi_N\}$ be a basis of S_h. Then equation (8.61) is equivalent to

$$a(u_h,\phi_i) - \langle l,\phi_i\rangle \quad i = 1,2,.....N.$$

With the formulation

$$u_h = \sum_{k=1}^{N} x_k\phi_k$$

we can define a linear system of equations

$$\sum_{k=1}^{N} a(\phi_k, \phi_i) x_k = \langle l, \phi_i \rangle \quad i = 1, 2, ...N,$$

that can be written in matrix form

$$Ax = b. \tag{8.62}$$

Here $A_{i,k} := a(\phi_k, \phi_i)$ and $b_i := \langle l, \phi_i \rangle$. The matrix A is positive definite

$$\begin{aligned} x^\top A x &= \sum_{i,k} x_i A_{ik} x_k = a\Big(\sum_k x_k \phi_k, \sum_i x_i \phi_i\Big) \\ &= a(u_h, u_h) \\ &\geq \alpha \|u_h\|_m^2, \end{aligned}$$

where $0 < \alpha \leq C$ and the bilinear form a is $\mathcal{H}^m$-elliptic

$$a(u,u) \geq \alpha \|u\|_m^2 \quad |a(u,v)| \leq C \|u\|_m \|v\|_m \quad \forall u, v \in \mathcal{H}.$$

The norm $\|\cdot\|$ is defined as $\|v\| = \sqrt{a(v,v)}$.

For the solution u_h of (8.61) the following stability condition is valid:

$$\|u_h\|_m \leq \frac{1}{\alpha} \|l\|.$$

The error estimation of a finite element approximation based on Céa-Lemma: Let the bilinear form a $\mathcal{H}$-elliptic and u respectively u_h the solution of the variational problem in $\mathcal{H}$ respectively S_h, then

$$\|u - u_h\|_m \leq \frac{C}{\alpha} \inf_{v_h \in S_h} \|l\|_m.$$

Céa-Lemma states that the error of any Galerkin approximation is only a constant factor (independent of h) higher than that of the best approximation of $u \in \mathcal{H}$.

8.5.3 Numerical Discretisation

The simplest discretisation is obtained by choosing a regular triangulation of the image domain Ω and attaching to each pixel position a piecewise linear basis function $\phi(x_1, x_2)$, as illustrated in figure 8.10.
Indexing each pixel position (k, l) by $1, 2, \ldots, N$ we thus have

$$u_1(x_1, x_2) = \sum_{i=1}^{N} u_i \phi_i(x_1, x_2) .$$

Hence, each of the functions u_1, u_2 is represented by N real variables. Hence:

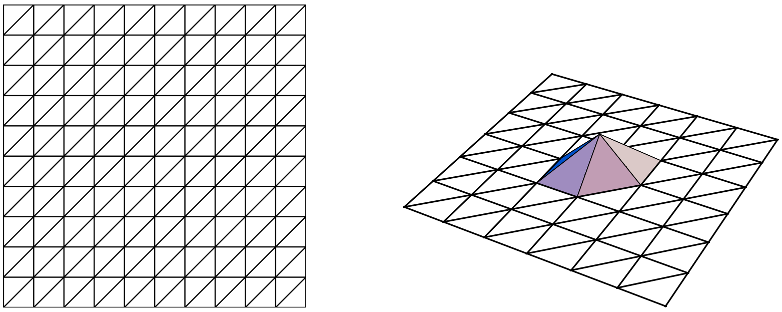


Figure 8.10: Left: Uniform triangulation of the image domain Ω. **Right:** Basis function $\phi_i(x_1, x_2)$ attached to a single pixel position i.

$$A \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = b. \tag{8.63}$$

The $2N \times 2N$–Matrix A factorises into

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{12}^\top & A_{22} \end{pmatrix},$$

where:

$$\begin{aligned}
(A_{11})_{k,l} &= a\big((\phi_k, 0)^\top, (\phi_l, 0)^\top\big) \\
(A_{12})_{k,l} &= a\big((\phi_k, 0)^\top, (0, \phi_l)^\top\big) \\
(A_{22})_{k,l} &= a\big((0, \phi_k)^\top, (0, \phi_l)^\top\big) \ .
\end{aligned}$$

Analogously, the $2N$–vector b factorises into $b = (b_1^\top, b_2^\top)^\top$ where:

$$\begin{aligned}
(b_1)_k &= b\big((\phi_k, 0)^\top\big) \\
(b_2)_k &= b\big((0, \phi_k)^\top\big) \ .
\end{aligned}$$

The linear system (8.63) is sparse and positive definite. Thus u_1, u_2 can be conveniently computed by some corresponding iterative solver [59]. For the numerical solution of the linear system (8.62) we refer to the next chapter 9.

9

Numerical Solvers

9.1 Conjugate Gradient Method

In this section we consider an efficient iterative method for the numerical solution of large systems of linear equations. Hestenes and Stiefel [65] published in 1952 the conjugate gradient (CG) method that is suitable for a class of problems which have a symmetric positive definite structure. The solution $\tilde{x}$ of the linear system

$$Ax = b \tag{9.1}$$

with the symmetric and positive definite matrix A is known to be the unique minimum of the convex quadratic function

$$f(x) = \frac{1}{2} x^\top A x - b^\top x. \tag{9.2}$$

The CG method starts from the initial $x^0 \in \mathbb{R}^n$ and subsequently provides a sequence of vectors $x^{(k)}$ which correspond to objective values $f(x^{(k)})$ that decrease monotonically.

The sequence $x^{(k)}$ converges thereby to the desired solution. The iterative update $x^{(k+1)}$ is obtained by using the estimate $x^{(k)}$ and going into the search direction given by the vector $p^{(k)}$:

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} p^{(k)}, \tag{9.3}$$

The residual vector is computed by

$$r^{(k)} = b - A x^{(k)} \tag{9.4}$$

which is used in the step length computation determined by the expression:

$$\alpha^{(k)} = \frac{r^{(k)\top}\, p^{(k)}}{p^{(k)\top}\, A p^{(k)}}. \tag{9.5}$$

Then

$$r^{(k+1)} = r^{(k)} - \alpha^{(k)} A p^{(k)} \tag{9.6}$$

is used in the update of the search direction:

$$p^{(k+1)} = r^{(k+1)} + \beta^{(k)} p^{(k)}, \tag{9.7}$$

where the scalar step length $\beta^{(k)}$ is computed as

$$\beta^{(k)} = \frac{r^{(k+1)\top}\, r^{(k+1)}}{r^{(k)\top}\, r^{(k)}}. \tag{9.8}$$

The new vector $x^{(k+1)}$ represents the minimum of the function $f(x)$ in the subspace spanned by the search directions $p^{(i)}, i = 0, 1, .., k$. This non-zero vectors satisfies the equations

$$p^{(i)\top} A p^{(j)} = 0, \quad \forall i \neq j. \tag{9.9}$$

and is known as conjugacy property.

In accordance with our experience the convergence rate of the conjugate gradient method for our problems can be improved using the preconditioned conjugate gradient method that is summarised in the following section.

9.1.1 Preconditioned Conjugate Gradient Method

The basic idea of the Preconditioned Conjugate Gradient method (PCG) is that the solution of (9.1) is the same as the solution of

$$P\ Ax = P\ b \tag{9.10}$$

for a non-singular matrix P. Using a linear transformation $\hat{x} = P^{\frac{1}{2}} x$ the objective function (9.2) becomes

$$f(\hat{x}) = \frac{1}{2} \hat{x}^\top (P^{-\frac{1}{2}} A P^{-\frac{1}{2}}) \hat{x} - (P^{-\frac{1}{2}} \hat{x})^\top b. \tag{9.11}$$

Like in the CG method the function $f(\hat{x})$ is minimised to obtain the solution of the linear equation. The number of conjugate steps can be reduced by choosing a non-singular matrix P such that the eigenvalues of $P^{-\frac{1}{2}} A P^{-\frac{1}{2}}$ are clustered more closely. A simple and often used standard preconditioner is the matrix $diag(A)$. This is the diagonal matrix that is formed if all off-diagonal elements in A are set to zero. Other preconditioners can be found in [18]. Our implementation which makes use of the preconditioner $diag(A)$ is summarised in Algorithm 3.

Algorithm 3 Preconditioned CG Algorithm

```
Require: $x^0 \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$ symmetric positive definit, $b \in \mathbb{R}^n$.
  Set $k = 0$, $r^0 = b - Ax^0$, $q^0 = diag(A)^{-1}r^0$, $p^0 = q^0$.
  For a given tolerance $\epsilon$,

  while $\frac{\|r^{(k)}\|}{\|b\|} > \epsilon$ do

    $\alpha^{(k)} = \frac{r^{(k)\top} q^{(k)}}{p^{(k)\top} Ap^{(k)}}$
    $x^{(k+1)} = x^{(k)} + \alpha^{(k)}p^{(k)}$
    $r^{(k+1)} = r^{(k)} - \alpha^{(k)}A\, p^{(k)}$
    $q^{(k+1)} = diag(A)^{-1}r^{(k+1)}$
    $\beta^{(k)} = \frac{r^{(k+1)\top} q^{(k+1)}}{r^{(k)\top} q^{(k)}}$
    $p^{(k+1)} = q^{(k+1)} + \beta^{(k)}\, p^{(k)}$
    $k = k + 1$

  end while

  $x^{(k)} \approx A^{-1}b$.
```

9.2 Coarse-to-Fine Approach and Iterative Registration

The accuracy of motion estimation critically depends on the magnitude of image motion. It is known that, depending on the spatial image frequency, very large motions even may cause aliasing along the time frequency axis. As a remedy, we first compute a coarse velocity field by using only low spatial frequency components. Then "undoing" the motion roughly stabilised the position of the image over time. Next, the higher frequency subbands are used to estimate the motion on the warped sequence. Combining the resulting correction term with the previously computed estimate results in a refined velocity estimate.

We exemplarily formulate the coarse to fine approach for the following energy functional

$$J(u) = \frac{1}{2}\int\limits_{\Omega} \Big(\nabla I^\top u + \partial_t I\Big)^2 dx \tag{9.12}$$

which is an essential part of some of the equations we were concerned with in this thesis, like (3.4) and (4.5).

Here u denotes the velocity field we are interested in, u_{old} represents the current estimate of the velocity field. The term $\partial_t I_{warp}$ is the temporal derivative which is computed as the difference between the second image - the one warped with u_{old} - and the first image. Then energy functional 9.12 can be reformulated as

$$J(u) = \frac{1}{2}\int\limits_{\Omega} \Big(\nabla I^\top (u - u_{old}) + \partial_t I_{warp}\Big)^2 dx. \tag{9.13}$$

Note that the motion over the image boundary Γ prevents the computation of the spatial and temporal gradients of the warped image I_{warp} at a specific location. In order to avoid error-prone filling-in heuristics we have chosen to omit the evaluation of the data term at these particular locations. The computation is repeated for every resolution level. We choose a zero velocity field as an initialisation.

9.3 Multigrid Methods

Multigrid methods [19, 59, 18, 45] have led to the implementation of very fast solvers for linear systems of equations and can cope with a large number of unknown variables. We exploit such multigrid methods to implement a solver that was appropriate to solve our linear systems that arise from the elliptic finite element discretisation of our problems.
In the following we sketch the algorithm we used to implement the solver (some details are denoted below):

Algorithm 4 Two-Grid Iteration

Pre-Smoothing: perform n Gauss-Seidel sweeps (9.15). Start with an initial estimate of $\tilde{x}_h$ and compute

$$d_h = A_h \tilde{x}_h - f_h, \tag{9.14}$$

where h denotes the mesh size on an uniform grid.
Coarse-Grid Correction: solve equation (9.16)-(9.18) to obtain the new value $\tilde{x}_h^{new}$
Post-Smoothing: perform m Gauss-Seidel sweeps (9.15) starting with $\tilde{x}_h^{new}$.

Gauss-Seidel Method

To obtain the n Gauss-Seidel sweeps for the mesh points 1,....,N we compute the following update

$$x_i = \frac{\left(\sum_{j=1, j\neq i}^{N} A_{ij} x_j - f_i\right)}{L_{ii}}, \quad i, ..., N. \tag{9.15}$$

Note that in this computation the updated elements of x are used as soon as they available.

Coarse-Grid Correction

In this part of the algorithm $\tilde{x}_h$ represents an estimate for the solution of the linear system. With that the deviation d_h from the desired solution is computed:

$$d_h = A_h \tilde{x}_h - f_h. \tag{9.16}$$

Here h denotes the mesh size of the underlying uniform grid. Since A_h is linear, the value d_h satisfies

$$A_h e_h = -d_h, \tag{9.17}$$

where e_h determines the difference between the estimated $\tilde{x}_h$ and the true x_h. To obtain the deviation at a coarser resolution d_{2h} is computed on the grid with the doubled mesh size $2h$:

$$d_{2h} = R d_h,$$

The matrix R is known as restriction operator and depends on the structure of the finite elements that are used for the discretisation [18].
Then the error e is computed on the coarse grid (compare (9.17)):

$$A_{2h} e_{2h} = -d_{2h}.$$

One can interpolate the error e_{2h} onto the finer grid e_h by the following projection

$$e_h = P e_{2h},$$

where P represents the projection operator, which depend again on the selected finite element discretisation. With that an updated approximation of the solution $(\tilde{x}_h^{new})$ is computed:

$$\tilde{x}_h^{new} = \tilde{x}_h + e_h. \tag{9.18}$$

We remark that this procedure usually reduce the occurrence of low-frequency errors.

10

Conclusion

10.1 Summary

In this chapter we summarise the essential points of this thesis. Based on our motivation - to transfer fluid flow estimation techniques into the field of image sequence processing - the main aim of this thesis was to incorporate physical prior knowledge in a variational framework for the motion analysis of image sequences. We introduced two new variational approaches for motion estimation of image sequences using physically motivated prior knowledge and control based interpretations. Furthermore we proposed a new optical flow based inpainting approach for video sequences.

One can determine the below listed limitations in current motion estimation and video inpainting methods. The incorporation of prior knowledge based on physical equations allowed to overcome these issues and represent a motivation for our approaches:

- Regularisation in spatio-temporal context is performed by evaluating the entire spatio-temporal image volume in a batch processing mode.
- Static view of image motion processing: The time-dependent optical flow is computed for a fixed point in time. In contrast to the "dynamic" viewpoint of the motion, which describes the fact that the image scene is in motion and that the positions of image intensities change with time.
- Spatio-temporal filtering: Motion estimation and video reconstruction is done regardless of the fact that moving structures usually exhibit some inertia in its most rudimentary form.

To develop a new method which is able to cope with these limitations we choose a framework based on variational methods. In chapter 2 we started with standard variational optical flow approaches, where the optical flow fields were globally estimated over the entire image domain. We adapted in chapter 3 the image-driven approach of Nagel (2.15) to incorporate a matching term that serves as a new regulariser which penalises the deviation from the predicted velocity values, obtained by the solution of the Burgers equation. The key features of the resulting method are provided below:

- **Physically Plausible Prior Knowledge:** We presented a novel PDE-based representation of image motion exploiting the most elementary prior knowledge available in connection with image motion computation which can be informally expressed as “structures do not jump”. The physics underlying any motion phenomena embody some inertia leading to smooth changes of velocities. Therefore, we exploited the aforementioned elementary knowledge by incorporating the Burgers equation providing the inertia behaviour in connection with motion computation.

- **Dynamic Image Motion:** We transferred the physically description of dynamic fluid motion [1] into the field of image motion computation. In particular, the dynamical behaviour is expressed by a physical partial differential equation - the nonlinear Burgers equation. We incorporate this physical equation within our variational formulation in order to mimic the dynamical behaviour of the fluid motion, by considering the scene as a rigid ”fictive fluid”, where its motion is described by the material derivative of the velocity with respect to time. Observed image measurements force the velocity into the appropriate motion direction.

- **Spatio-Temporally Filtering:** We pointed out in chapter 3 that our model exhibit, in a distributed-parameter fashion [2], properties analogous to the dynamic filter for the most elementary polynomial kinematical model of a point feature (cf. [6]). This means that velocities of an estimated flow field do not change in the absence of image measurements and if the spatial context does not indicate otherwise.

- **Online Processing:** Our approach performs a spatio-temporal regularisation in a recursive online fashion, meaning that it can already start before the full sequence has been recorded. The solutions of our approach are recursively computed based on the fact that velocities are transported by itself over time.

- **Control based Interpretation:** If image measurements indicate changes of the current velocity distribution, fictive forces modify the system state accordingly. The presence of such forces may serve as an attentional mechanism notifying a higher-level processing stage about unexpected motion events. In this context, we were tempted to point out a potential relevance of our approach for related models of visual perception [91].

---

[1] Also known as Lagrangian description of motion, where we consider changes, which occur as you follow a fluid particle over time. We observe the ”trajectory” view of the fluid expressed as total derivative of the velocity with respect to time, the so called *material derivative*.

[2] Distributed parameter system theory is concerned with the dynamic behaviour of processes distributed in space as well as evolving in time [51]. The states of these systems are described and governed by partial differential equations (PDEs), rather than ordinary differential equations as for lumped systems.

However, as we are not experts in this field, we presented only a few speculative remarks.

Control based Optical Flow Approach

In real-world applications however, we are often concerned with long image sequences which allows the utilisation of the temporal coherency as well as physical assumptions of the motion. In chapter 4 we presented a control based optical flow approach for image motion estimation, which results in a global temporal regularisation of the flow field. The advantage of a variational formulation of a control problem is a clear and similar mathematical formalism for all model assumptions. The solutions, obtained by minimising the energy functional, are optimal with respect to the specified assumptions. Optical flow fields are forced to vector fields which satisfy the Burgers equation. Control variables are included that allow to adjust the image motion in such a way that it fits to the apparent velocities of the moving objects in a given image sequence. Our presented approach is an attempt to adopt respective methods from the field of applied mathematics in a new form to image sequence processing and to evaluate the capability of this approach. The application of optimal control techniques to image motion estimation, as presented in this work, is however, novel and explorative.

Dynamic Image Motion based Video Inpainting Approach

In chapter 6 we adapted our optical flow approach from chapter 3 to incorporate spatial temporal coherency within a new PDE based video inpainting approach. Image information is transported in a dynamical way and preserves structural information originating from previous image frames.

10.2 Open Problems and Further Work

In this section we summarise several ideas, which are worth to be investigated in the future and might improve and expand the techniques and approaches used in this thesis.

- **Regularisation Term:** The regularisation term in the variational framework is responsible for the incorporation of priors like the smoothness constraint. In our approaches it is also used to incorporate physical prior knowledge suitable in particular for rigid and non-rigid motion. To improve the robustness of our approach with respect to noise the use of Total-Variation regularisation techniques could result in more accurate motion fields. TV-regularisation represents a denoising technique permitting discontinuous solutions [114].

- **Occlusions and Motion Boundaries:** Some methods incorporate occlusion handling either as a post processing step or by a direct modelling within the motion estimation framework. The results published in [72, 1] show that further improvements in this problem field are desired. However, the direct handling of occlusions and motion boundaries was excluded from our modelling framework. We think that methods that include several motion layers for separated motion estimations might be able to cope with this problem. Therefore, in further work one should address the problem to estimate different motion layers and how motion layers can be exploit to model the presence of objects and occlusions.

- **Receding Horizon Control Formulation:** In the current implementation our control approach needs the full image sequence information before the sequence can be processed. Innovative approaches to overcome this drawback are receding horizon control formulations [73]. These techniques use a "moving window" implementation to solve approximately the optimal control problem in an online computation over the entire sequence. This means that the current control at a specific state x and time t is obtained by determining online the optimal control f over the interval $[t, t+T]$, which is repeated until a new state update has been obtained. Therefore one should explore the possibilities to reformulate the current approach into a receding horizon control formulation.
- **Video Editing:** An application that could arise from our research - belonging to video editing - is the automatic removal of moving objects within an image sequence. However, this requires a reliable segmentation of the moving inpaint area [56] and would mean that one has to be able to detect the motion boundaries quite accurately.
- **Numerical Aspects:** We used in our discretisation the assumption of uniform mesh grids. Therefore, one should investigate the potential of the application of non-uniform grids, which could adapt to more complex geometries. In regions where discontinuities or shock formations occur, the mesh form and size should automatically adapt to model these regions more accurately. In contrast the size of the specific mesh should increase with an increase of the dominant spatial and temporal dimensions.
  In order to further optimise the computation speed, the usability of parallelisation techniques and problem specific solvers should be investigated (cf. [79, 80, 146]).

A

Appendix

In this appendix we collected the definitions, derivations and computational expensive formulas from chapter 4.

A.1 Derivation of the Optimality System

In this section we derive the first-order necessary conditions for the Lagrangian functional (4.5) from section 4.2.1, which leads us to the optimality system ($4.7a$ - $4.7c$) for the determination of the optimal states u and control f (cf. [58]).

Setting the first variation of the Lagrangian functional (4.5) with respect to the Lagrangian multiplier w equal to zero yields

$$\begin{aligned}
\frac{\partial L(w+\varepsilon\tilde{w})}{\partial\varepsilon}\,|_{\varepsilon=0} &= \frac{\partial}{\partial\varepsilon}\int\limits_{\Omega\times[0,T]} \Big\{\frac{1}{2}(\partial_t I + u\cdot\nabla I)^2 + \frac{1}{2}\alpha(|\nabla u_1|^2 + |\nabla u_2|^2) \\
&\quad +\frac{1}{2}\beta|f|^2 - (w+\varepsilon\tilde{w})\cdot(\partial_t u + (u\cdot\nabla)u - f)\Big\}\,dxdt\,|_{\varepsilon=0} \\
&= -\int\limits_{\Omega\times[0,T]} \tilde{w}\cdot(\partial_t u + (u\cdot\nabla)u - f)\,dxdt = 0.
\end{aligned}$$

Since the variation $\tilde{w}$ in the Lagrangian multiplier w is arbitrary, we recover the constraint or *state equation*:

$$\partial_t u + (u\cdot\nabla)u - f = 0 \quad \text{in } \Omega\times[0,T], \tag{A.1}$$

with von Neumann boundary conditions $\partial_n u = 0$ on $\Gamma\times[0,T]$. Setting the first variation of the Lagrangian functional (4.5) with respect to the velocity state u equal to zero yields

$$\begin{aligned}\frac{\partial L(u+\varepsilon\tilde{u})}{\partial\varepsilon}\Big|_{\varepsilon=0} &= \int\limits_{\Omega\times[0,T]} \Big\{\tilde{u}\cdot\nabla I(\partial_t I + u\cdot\nabla I) + \alpha(\nabla u_1\cdot\nabla\tilde{u}_1 + \nabla u_2\cdot\nabla\tilde{u}_2) \\ &\quad -w\cdot(\partial_t\tilde{u} + (u\cdot\nabla)\tilde{u} + (\tilde{u}\cdot\nabla)u)\Big\}\, dxdt.\end{aligned}$$

Using integration by parts we obtain the following expression

$$\begin{aligned}\frac{\partial L(u+\varepsilon\tilde{u})}{\partial\varepsilon}\Big|_{\varepsilon=0} &= \int\limits_{\Omega\times[0,T]} (\partial_t I + u\cdot\nabla I)\tilde{u}\cdot\nabla I\; dxdt \\ &\quad - \int\limits_{\Omega\times[0,T]} \alpha(\Delta u_1\tilde{u}_1 + \Delta u_2\tilde{u}_2)\; dxdt + \int\limits_{\Gamma\times[0,T]} (\partial_n u_1\tilde{u}_1 + \partial_n u_2\tilde{u}_2)\; d\Gamma dt \\ &\quad + \int\limits_{\Omega\times[0,T]} \partial_t w\cdot\tilde{u}\; dxdt - \int\limits_{\Gamma\times[0,T]} w\cdot\tilde{u}n_t\; d\Gamma dt \\ &\quad -\Big\{-\int\limits_{\Omega\times[0,T]} \partial_{x_1}(w_1u_1)\tilde{u}_1\; dxdt + \int\limits_{\Gamma\times[0,T]} w_1u_1\tilde{u}_1n_{x_1}\; d\Gamma dt \\ &\quad - \int\limits_{\Omega\times[0,T]} \partial_{x_2}(w_1u_2)\tilde{u}_1\; dxdt + \int\limits_{\Gamma\times[0,T]} w_1u_2\tilde{u}_1n_{x_2}\; d\Gamma dt \\ &\quad - \int\limits_{\Omega\times[0,T]} \partial_{x_1}(w_2u_1)\tilde{u}_2\; dxdt + \int\limits_{\Gamma\times[0,T]} w_2u_1\tilde{u}_2n_{x_1}\; d\Gamma dt \\ &\quad - \int\limits_{\Omega\times[0,T]} \partial_{x_2}(w_2u_2)\tilde{u}_2\; dxdt + \int\limits_{\Gamma\times[0,T]} w_2u_2\tilde{u}_2n_{x_2}\; d\Gamma dt \\ &\quad + \int\limits_{\Omega\times[0,T]} (w_1\tilde{u}_1\partial_{x_1}u_1 + w_1\tilde{u}_2\partial_{x_2}u_1 + w_2\tilde{u}_1\partial_{x_1}u_2 + w_2\tilde{u}_2\partial_{x_2}u_2)\; dxdt\Big\}.\end{aligned}$$

The component-wise notation leads us to the vector notation:

$$\begin{aligned}\frac{\partial L(u+\varepsilon\tilde{u})}{\partial\varepsilon}\Big|_{\varepsilon=0} &= \int\limits_{\Omega\times[0,T]} \Big\{\nabla I(\partial_t I + u\cdot\nabla I) - \alpha\Delta u \\ &\qquad +\partial_t w + (u\cdot\nabla)w + w\nabla\cdot u - (\nabla U)^\top w\Big\}\cdot\tilde{u}\; dxdt \\ &\quad + \int\limits_{\Gamma\times[0,T]} \Big\{\partial_n u - w(u\cdot n) - wn_t\Big\}\cdot\tilde{u}\; d\Gamma dt,\end{aligned}$$

where

$$(\nabla U) = \begin{pmatrix} \partial_{x_1}u_1 & \partial_{x_2}u_1 \\ \partial_{x_1}u_2 & \partial_{x_2}u_2 \end{pmatrix}$$

denotes the Jacobian matrix of u [32]. Since the variation $\tilde{u}$ in the Lagrangian multiplier u is arbitrary, we recover the *adjoint equation*:

$$= \begin{cases} \nabla I(\partial_t I + u \cdot \nabla I) - \alpha \Delta u & \\ +\partial_t w + (u \cdot \nabla) w + w \nabla \cdot u - (\nabla U)^\top w = 0 \ , & \text{in } \Omega \times [0,T], \\ & \\ \partial_n u - w(u \cdot n) - w n_t = 0 \ , & \text{on } \Gamma \times [0,T], \end{cases} \tag{A.2}$$

with $\partial_n u = 0$ this is satisfied with $w = 0$ on the boundary $\Gamma \times [0,T]$.

With this derivation, the adjoint system is posed as a backward in time problem and the adjoint velocity satisfies the terminal condition at

$$w|_{t=T} = 0. \tag{A.3}$$

And finally, setting the first variation of the Lagrangian functional (4.5) with respect to the control f equal to zero yields

$$\frac{\partial L(f + \varepsilon \tilde{f})}{\partial \varepsilon} |_{\varepsilon=0} \quad = \int\limits_{\Omega \times [0,T]} \beta(f + \varepsilon \tilde{f}) \cdot \tilde{f} + w \cdot \tilde{f} \; dxdt \, |_{\varepsilon=0}.$$

Since the variation $\tilde{f}$ in the Lagrangian multiplier f is arbitrary, we recover the *optimality condition*:

$$\beta f + w = 0 \ , \quad \text{in } \Omega \times [0,T], \tag{A.4}$$

with Dirichlet boundary conditions $f = 0$ on $\Gamma \times [0,T]$. With the terminal condition $w|_{t=T} = 0$ results for the the control function of the optimality condition (A.4) the terminal condition

$$f|_{t=T} = 0 \tag{A.5}$$

(cf. [58]).

A.2 The Gradient of the Functional Through Sensitivities

To determined the gradient of the functional (4.3) we need to determine the total derivative of the functional with respect to the control (cf. [58]). Therefore, we define, that the state equation (4.7a) is solved to determine the state u as a function of the control f so that the functional $J(u,f) = J(u(f),f)$ is a function of only the control f. As a consequence, we apply the chain rule to $J(u(f),f)$ and obtain

$$\partial_f J = \frac{\partial J}{\partial u} \frac{du}{df} + \frac{\partial J}{\partial f} \tag{A.6}$$

the formula for the total derivative of the functional (4.3). Nevertheless, we need to specify the term $\frac{du}{df}$ for the determination of the gradient of the functional. This term defines the so-called *sensitivities* and describes what changes are effected in the state when the control variables are changed. If we change the distributed control f to $f + \varepsilon\tilde{f}$, with $\tilde{f}$ arbitrary this induces the variation of the state from u to $u + \varepsilon u_f$. Then from the state equation (4.7a) we have that the corresponding change u_f in the state must satisfy

$$\begin{aligned}
\partial_t(u+\varepsilon u_f) + ((u+\varepsilon u_f)\cdot\nabla)(u+\varepsilon u_f) &= f+\varepsilon\tilde{f} \quad \text{in } \Omega\times[0,T], && (\text{A.8}a)\\
\partial_n(u+\varepsilon u_f) &= 0 \quad \text{on } \Gamma\times[0,T]. && (\text{A.8}b)
\end{aligned}$$

We reformulate equation (A.8a) into a component-wise notation:

$$\begin{cases}
\partial_t u_1 + \varepsilon\partial_t u_{1_f} + u_1\partial_{x_1}u_1 + u_1\varepsilon\partial_{x_1}u_{1_f} + \varepsilon u_{1_f}\partial_{x_1}u_1 + \varepsilon u_{1_f}\varepsilon\partial_{x_1}u_{1_f} & \\
\qquad +u_2\partial_{x_2}u_1 + u_2\varepsilon\partial_{x_2}u_{1_f} + \varepsilon u_{2_f}\partial_{x_2}u_1 + \varepsilon u_{2_f}\varepsilon\partial_{x_2}u_{1_f} & = f_1 + \varepsilon\tilde{f}_1,\\
\partial_t u_2 + \varepsilon\partial_t u_{2_f} + u_1\partial_{x_1}u_2 + u_1\varepsilon\partial_{x_1}u_{2_f} + \varepsilon u_{1_f}\partial_{x_1}u_2 + \varepsilon u_{1_f}\varepsilon\partial_{x_1}u_{2_f} & \\
\qquad +u_2\partial_{x_2}u_2 + u_2\varepsilon\partial_{x_2}u_{2_f} + \varepsilon u_{2_f}\partial_{x_2}u_2 + \varepsilon u_{2_f}\varepsilon\partial_{x_2}u_{2_f} & = f_2 + \varepsilon\tilde{f}_2.
\end{cases}$$

If we let $\varepsilon \to 0$ we have

$$\begin{cases}
\partial_t u_{1_f} + u_1\partial_{x_1}u_{1_f} + u_{1_f}\partial_{x_1}u_1 + u_2\partial_{x_2}u_{1_f} + u_{2_f}\partial_{x_2}u_1 &= \tilde{f}_1,\\
\partial_t u_{2_f} + u_1\partial_{x_1}u_{2_f} + u_{1_f}\partial_{x_1}u_2 + u_2\partial_{x_2}u_{2_f} + u_{2_f}\partial_{x_2}u_2 &= \tilde{f}_2.
\end{cases}$$

Finally, by using the analogous procedure for the boundary conditions we get the *sensitivity equation* by

$$\begin{cases}
\partial_t u_f + (u_f\cdot\nabla)u + (u\cdot\nabla)u_f = \tilde{f} & \text{in } \Omega\times[0,T],\\
\partial_n u_f = 0 & \text{on } \Gamma\times[0,T].
\end{cases} \qquad (\text{A.9})$$

From the sensitivity equation (A.9) one can see that an infinitesimal change of the control in the direction of the control induces an corresponding infinitesimal change in the direction of the state.

In a next step we have to determine the infinitesimal change of the objective functional $J(u, f)$ (4.3) (infinitesimal change of the objective functional J is denoted by $\partial_f J$) effected by an infinitesimal change in the direction $\tilde{f}$ in the control f. We will keep both the explicit dependence of J on f and the implicit dependence through the state u:

$$\begin{aligned}
\langle\partial_f J, \tilde{f}\rangle &= \frac{\partial J(u+\varepsilon u_f, f+\varepsilon\tilde{f})}{\partial\varepsilon}\Big|_{\varepsilon=0}\\
&= \int\limits_{\Omega\times[0,T]} \left\{(\partial_t I + u\cdot\nabla I)\nabla I\cdot u_f - \alpha\Delta u\cdot u_f + \beta f\cdot\tilde{f}\right\}\, dxdt.
\end{aligned}$$

The substitution of the adjoint equation (4.7b) yields

$$\langle \partial_f J, \tilde{f} \rangle = \int\limits_{\Omega\times[0,T]} \Big\{ -(\partial_t w + (u\cdot\nabla)w + w\nabla\cdot u - (\nabla U)^\top w)\cdot u_f + \beta f\cdot\tilde{f} \Big\}\, dxdt.$$

Using integration by parts we obtain the following expression

$$\begin{aligned}
\langle \partial_f J, \tilde{f} \rangle = & \int\limits_{\Omega\times[0,T]} \Big(\partial_t u_{1_f} w_1 + \partial_t u_{2_f} w_2 \Big)\, dxdt - \int\limits_{\Gamma\times[0,T]} \Big(u_{1_f} w_1 n_t + u_{2_f} w_2 n_t \Big)\, d\Gamma dt \\
& + \int\limits_{\Omega\times[0,T]} \Big(\partial_{x_1}(u_1 u_{1_f}) w_1 + \partial_{x_2}(u_2 u_{1_f}) w_1 + \partial_{x_1}(u_1 u_{2_f}) w_2 + \partial_{x_2}(u_2 u_{2_f}) w_2 \Big)\, dxdt \\
& - \int\limits_{\Gamma\times[0,T]} \Big(u_1 u_{1_f} w_1 n_{x_1} + u_2 u_{1_f} w_1 n_{x_2} + u_1 u_{2_f} w_2 n_{x_1} + u_2 u_{2_f} w_2 n_{x_2} \Big)\, d\Gamma dt \\
& - \int\limits_{\Omega\times[0,T]} \Big(w_1 \partial_{x_1} u_1 u_{1_f} + w_1 \partial_{x_2} u_2 u_{1_f} + w_2 \partial_{x_1} u_1 u_{2_f} + w_2 \partial_{x_2} u_2 u_{2_f} \Big)\, dxdt \\
& + \int\limits_{\Omega\times[0,T]} \Big(\partial_{x_1} u_1 w_1 u_{1_f} + \partial_{x_1} u_2 w_2 u_{1_f} + \partial_{x_2} u_1 w_1 u_{2_f} + \partial_{x_2} u_2 w_2 u_{2_f} \Big)\, dxdt \\
& + \int\limits_{\Omega\times[0,T]} \beta f\cdot\tilde{f}\, dxdt.
\end{aligned}$$

By simplification

$$\begin{aligned}
\langle \partial_f J, \tilde{f} \rangle = & \int\limits_{\Omega\times[0,T]} \Big(\partial_t u_{1_f} w_1 + \partial_t u_{2_f} w_2 \Big)\, dxdt - \int\limits_{\Gamma\times[0,T]} \Big(u_{1_f} w_1 n_t + u_{2_f} w_2 n_t \Big)\, d\Gamma dt \\
& + \int\limits_{\Omega\times[0,T]} \Big(u_{1_f} \partial_{x_1} u_1 w_1 + u_{2_f} \partial_{x_2} u_1 w_1 + u_{1_f} \partial_{x_1} u_2 w_2 + u_{2_f} \partial_{x_2} u_2 w_2 \Big)\, dxdt \\
& + \int\limits_{\Omega\times[0,T]} \Big(u_1 \partial_{x_1} u_{1_f} w_1 + u_2 \partial_{x_2} u_{1_f} w_1 + u_1 \partial_{x_1} u_{2_f} w_2 + u_2 \partial_{x_2} u_{2_f} w_2 \Big)\, dxdt \\
& + \int\limits_{\Omega\times[0,T]} \beta f\cdot\tilde{f}\, dxdt. \\
& - \int\limits_{\Gamma\times[0,T]} \Big(u_1 u_{1_f} w_1 n_{x_1} + u_2 u_{1_f} w_1 n_{x_2} + u_1 u_{2_f} w_2 n_{x_1} + u_2 u_{2_f} w_2 n_{x_2} \Big)\, d\Gamma dt
\end{aligned}$$

the component-wise notation leads us to the vector notation:

$$\begin{aligned}\langle \partial_f J, \tilde{f}\rangle &= \int\limits_{\Omega\times[0,T]} \Big\{\Big(\partial_t u_f + (u_f\cdot\nabla)u + (u\cdot\nabla)u_f\Big)\cdot w + \beta f\cdot\tilde{f}\Big\}\; dxdt \\ &\quad - \int\limits_{\Gamma\times[0,T]} \Big(u_f(u\cdot n + n_t)\Big)\cdot w\; d\Gamma.\end{aligned} \tag{A.10}$$

In this expression, the boundary integral ($\int\limits_{\Gamma\times[0,T]} ...d\Gamma\times[0,T]$) is always equal to zero as $w=0$ is valid on the boundary (cf. (A.2) with $\partial_n u = 0$ on $\Gamma\times[0,T]$ leads to $w=0$). Then the substitution of the sensitivity equation (A.9) into the equation (A.10) yields

$$\begin{aligned}\langle \partial_f J, \tilde{f}\rangle &= \int\limits_{\Omega\times[0,T]} \Big(\tilde{f}\cdot w + \beta f\cdot\tilde{f}\Big)\; dxdt \\ &= \int\limits_{\Omega\times[0,T]} (w+\beta f)\cdot\tilde{f}\; dxdt.\end{aligned}$$

This results in an equation for the gradient of our objective functional (4.3)

$$\partial_f J = w + \beta f. \tag{A.11}$$

The full previous computation shows that we can apply a gradient algorithm in order to solve the adjoint equation (4.7b) with an initial condition $u=0$ with a following update of the control variables by using the gradient of the objective functional (A.11). With the updated controls we solve the state equation (4.7a) and proceed to the next iteration. Consequently, after the convergence of the algorithm the optimality condition (4.7c) is satisfied, too.

Bibliography

[1] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sànchez. Symmetrical dense optical flow estimation with occlusion detection. In *ECCV*, pages 721–735. Springer, 2002.

[2] L. Alvarez, J. Esclarìn, M. Lefebure, and J. Sànchez. A PDE model for computing the optical flow. In *Proceedings of CEDYA XVI, Universidad de Las Palmas de Gran Canaria*, pages 1349–1356, 1999.

[3] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, volume 147 of *Applied Mathematical Sciences*. Springer, 2006.

[4] G. Aubert and L. A. Vese. A variational method in image recovery. *SIAM J. Numer. Anal.*, 34:1948–1979, 1997.

[5] C. Ballester, M. Bertalmío, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing*, 10(8):1200–1211, 2001.

[6] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association.* Academic Press, 1988.

[7] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[8] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes. Computing large deformation metric mappings via geodesic flows. *International Journal of Computer Vision*, 61(2):139–157, 2005.

[9] M. Bertalmío, A. L. Bertozzi, and G. Sapiro. Navier-Stokes, fluid dynamics, and image and video inpainting. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVRP*, volume 1, pages 355–362, 2001.

[10] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *SIGGRAPH '00: Proceedings of the 27th annual conference on computer graphics and interactive techniques*, pages 417–424, New York, 2000. ACM Press, Addison-Wesley Publishing Co.

[11] M. Bertalmío, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, 2003.

[12] M. Bertero, T. Poggio, and V. Torre. Ill-posed problems in early vision. Technical report, Cambridge, MA, USA, 1988.

[13] G. Birkhoff and C. de Boor. Piecewise polynomial interpolation and approximation. *Approximation of Functions*, pages 164–190, 1965.

[14] M. J. Black. *Robust Incremental Optical Flow.* PhD thesis, Yale University, Sep. 1992.

[15] M. J. Black, D. J. Fleet, and Y. Yacoob. Robustly estimating changes in image appearance. *Computer Vision and Image Understanding*, 78:8–31, 2000.

[16] A. Borzi, K. Ito, and K. Kunisch. Optimal control formulation for determining optical flow. *SIAM J. Sci. Comput.*, 24(3):818–847, 2002.

[17] D. Bosq. *Linear Processes in Function Spaces*, volume 149 of *Lect. Not. Statistics.* Springer, 2000.

[18] D. Braess. *Finite Elements. Theory, Fast Solver, and Applications in Solid Mechanics.* Springer, 1997.

[19] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977.

[20] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods.* Springer, New York, NY, USA, 1991.

[21] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Computer Vision - ECCV 2004*, pages 25–36, 2004.

[22] A. Bruhn and J. Weickert. Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 749–755, Washington, DC, USA, 2005. IEEE Computer Society.

[23] J. M. Burgers. A mathematical model illustrating the theory of turbulence. *Adv. Appl. Mech.*, 1:171–199, 1948.

[24] V. Caselles, J. M. Morel, and C. Sbert. An axiomatic approach to image interpolation. In *Proceedings of the 1997 International Conference on Image Processing (ICIP '97)*, volume 3, page 376, Washington, DC, USA, 1997. IEEE Computer Society.

[25] A. Chambolle and P. L. Lions. Image recovery via total variation minimization and related problems. *Numer. Math.*, 76:167–188, 1997.

[26] T. F. Chan, S. Kang, and J. Shen. Euler's elastica and curvature based inpaintings. *SIAM Journal on Applied Mathematics*, 63(2):564–592, 2002.

[27] T. F. Chan and J. Shen. Mathematical models for local non-texture inpaintings. *SIAM Journal on Applied Mathematics*, 62(3):1019–1043, 2001.

[28] T. F. Chan and J. Shen. Morphologically invariant PDE inpaintings. CAM Report 01-15, UCLA, 2001.

[29] T. F. Chan and J. Shen. Non-texture inpaintings by curvature-driven diffusions. *J. Visual Comm. Image Rep.*, 12(4):436–449, 2001.

[30] T. F. Chan and J. Shen. Variational restoration of non-flat image features: Models and algorithms. *SIAM Journal on Applied Mathematics*, 61(4):1338–1361, 2001.

[31] L. Chanas, J.P. Cocquerez, and J. Blanc-Talon. Numerical energy minimization for removing high luminous energy in dazzled images. In *Advanced Concepts for Intelligent Vision Systems (ACIVS99), Baden-Baden, Germany*, 2000.

[32] A. J. Chorin and J. E. Marsden. *A Mathematical Introduction to Fluid Mechanics.* Texts in Applied Mathematics. Springer, 1993.

[33] G. E. Christensen, R. D. Rabbit, and M. I. Miller. Deformable templates using large deformation kinematics. *IEEE Trans. Imag. Proc.*, 5(10):1435–1447, 1996.

[34] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North-Holland Publ. Comp., 1978.

[35] I. Cohen. Nonlinear variational method for optical flow computation. In *Proceedings of the 8th Scandinavian Conference on Image Analysis*, pages 523–530, Tromsø, Norway, 1993. IAPR.

[36] P. Colella and E. G. Puckett. *Modern Numerical Methods for Fluid Flow.* Lecture Notes, Department of Mechanical Engineering, University of California, Berkeley, CA, 1998. http://www.rzg.mpg.de/ bds/numerics/cfd-lectures.html.

[37] R. Courant, K. O. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100:32–74, 1982.

[38] R. Courant, E. Isaacson, and M. Rees. On the solution of nonlinear hyperbolic differential equations by finite differences. *Comm. Pure. Appl. Math.*, 5:243–255, 1952.

[39] A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based inpainting. volume 2, page 721, Los Alamitos, CA, USA, 2003. CVPR: IEEE Computer Society.

[40] R. Deriche, P. Kornprobst, and G. Aubert. Optical-flow estimation while preserving its discontinuities: A variational approach. In *ACCV*, pages 71–80, 1995.

[41] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[42] Q. Du, M. D. Gunzburger, and J. S. Peterson. Analysis and approximation of the Ginzburg-Landau model of superconductivity. *SIAM Rev.*, 34(1):54–81, 1992.

[43] B. Einfeldt. On Godunov-type methods for gas dynamics. *SIAM, J. Num. Anal.*, 25(2), 1988.

[44] G. Emile-Male. *The Restorer's Handbook of Easel Painting*. Van Nostrand Reinhold, New York, 1976.

[45] W. Enkelmann. Investigation of multigrid algorithms for the estimation of optical flow fields in image sequences. *Computer Vision, Graphics, and Image Processing*, 43:150–177, 1988.

[46] S. Esedoglu and J. Shen. Digital inpainting based on the Mumford Shah Euler image model. *European J. Appl. Math.*, 13:353–370, 2002.

[47] L. C. Evans. *Partial differential equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 1998.

[48] Guichard F. and Rudin L. Accurate estimation of discontinuous optical flow by minimizing divergence related functionals. *Proceedings of International Conference on Image Processing*, 1:497–500, 1996.

[49] C. A. J. Fletcher. *Computational Techniques for Fluid Dynamics (Vol.I)*. Springer, 1992.

[50] J. E. Fromm. A method for reducing dispersion in convective difference schemes. *Journal of Computational Physics*, 3(2):176–189, 1968.

[51] D. H. Gay and W. H. Ray. Identification and control of distributed parameter systems by means of the singular value decomposition. *Chemical Engineering Science*, 50(10):1519–1539, 1995.

[52] J. J. Gibson. *The Perception of the Visual World*. Houghton Mifflin, Boston, 1950.

[53] V. L. Ginzburg and L. D. Landau. On the theory of superconductivity. *Soviet Phys. JETP*, 20:1064, 1950.

[54] S. K. Godunov. A finite difference method for the computation of discontinuous solutions of the equation of fluid dynamics. *Mat. Sb.*, 47:357–393, 1959.

[55] H. Grossauer. A combined PDE and texture synthesis approach to inpainting. In *ECCV*, volume 2, pages 214–224, 2004.

[56] H. Grossauer. Inpainting of movies using optical flow. In *Mathematical Models for Registration and Applications to Medical Imaging*, volume 10, pages 151–162. Springer, 2006.

[57] H. Grossauer and O. Scherzer. Using the complex Ginzburg-Landau equation for digital inpainting in 2D and 3D. In *Scale-Space*, pages 225–236, 2003.

[58] M. Gunzburger. *Perspectives in Flow Control and Optimization.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002.

[59] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations*, volume 95 of *Applied Mathematical Sciences.* Springer, 1993.

[60] J. Hadamard. *Lectures on the Cauchy Problem in Linear Partial Differential Equations.* Yale University Press, New Haven, 1923.

[61] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.*, 49:357–393, 1983.

[62] H. W. Haussecker and D. J. Fleet. Computing optical flow with physical models of brightness variation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):661–673, 2001.

[63] J. Heel. Dynamic systems and motion vision. A.I. Memo No. 1037, MIT AI-Lab., 1988.

[64] J. Heel. Dynamic motion vision. In *Proc. Image Understanding Workshop*, pages 702–713. Morgan Kaufmann Publ. Inc., Palo Alto, 1989.

[65] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[66] C. Hirsch. *Numerical Computation of Internal and External Flows (Vol.I+II).* John Wiley & Sons, 2000.

[67] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

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

[69] T. Y. Hou and P. LeFloch. Why non-conservative schemes converge to the wrong solutions: Error analysis. *Math. of Comput.*, 62:497–530, 1994.

[70] W. Hundsdorfer and J. G. Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, volume 33 of *Series in Computational Mathematics*. 2003.

[71] H. Igehy and L. Pereira. Image replacement through texture synthesis. In *Proceedings of the 1997 International Conference on Image Processing (ICIP '97)*, volume 3, page 186, Washington, DC, USA, 1997. IEEE Computer Society.

[72] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4:324–335, 1993.

[73] K. Ito and K. Kunisch. Receding horizon optimal control for infinite dimensional systems. *ESAIM: Control, Optimisation and Calculus of Variations*, 8:741–760, 2002.

[74] B. Jähne. *Digitale Bildverarbeitung*. Springer, Berlin, 1997.

[75] R. Jain, R. Kasturi, and B. G. Schunck. *Machine Vision*. McGraw-Hill, Inc., 1995.

[76] J. Jost. *Partial Differential Equations*, volume 214 of *Graduate Texts in Mathematics*. Springer, 2007.

[77] G. Kanisza. *Organisation in Vision. Essays on Gestalt Perception*. Praeger, New York, 1979.

[78] S. Karni. Hybrid multifluid algorithms. Technical report 95-100, Courant Mathematics and Computing Laboratory, 1995.

[79] T. Kohlberger, C. Schnörr, A. Bruhn, and J. Weickert. Domain decomposition for parallel variational optical flow computation. In B. Michaelis and G. Krell, editors, *Pattern Recognition, Proc. 25th DAGM Symposium*, volume 2781, pages 196–203. Springer, 2003.

[80] T. Kohlberger, C. Schnörr, A. Bruhn, and J. Weickert. Parallel variational motion estimation by domain decomposition and cluster computing. In T. Pajdla and J. Matas, editors, *Proc. ECCV 2004*, volume 3024, pages 205–216. Springer, 2004.

[81] A. Kokaram. *Motion Picture Restoration*. Springer, 1998.

[82] A. Kokaram. Practical MCMC for missing data treatment in degraded video. In *Proceedings of European Conference on Computer Vision. Workshop on statistical methods in video processing, Copenhagen, Denmark*, pages 85–90, 2002.

[83] P. Kornprobst, R. Deriche, and G. Aubert. Image sequence restoration: A PDE based coupled method for image restoration and motion segmentation. In *ECCV98*, pages 548–562.

[84] A. Kumar, A. R. Tannenbaum, and G. J. Balas. Optical flow: A curve evolution approach. *IEEE Transactions on Image Processing*, 5(4):598–610, 1996.

[85] L. D. Landau and E. M. Lifschitz. *Hydrodynamik, Lehrbuch der Theoretischen Physik, Band VI.* Akademie-Verlag, Berlin, 1991.

[86] P. D. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure Appl. Math.*, 13:217–237, 1960.

[87] R. J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhäuser Verlag, 1990.

[88] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision (darpa). In *Proceedings of the 1981 DARPA Image Understanding Workshop*, pages 121–130, 1981.

[89] S. Masnou. Disocclusion: A variational approach using level lines. *IEEE Trans. Image Process.*, 11(2):6876, 2002.

[90] S. Masnou and J.M. Morel. Level lines based disocclusion. In *Proc. 1998 IEEE International Conference on Image Processing*, volume 3, pages 259–263, 1998.

[91] G. Mather, F. Verstraten, and S. Anstis. *The Motion Aftereffect: A Modern Perspective.* The MIT Press, 1998.

[92] L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236, 1989.

[93] B. McCane, K. Novins, D. Crannitch, and B. Galvin. On benchmarking optical flow. *Comput. Vis. Image Underst.*, 84(1):126–143, 2001.

[94] G. Moretti. The λ-scheme. *Computers and Fluids*, 7:191–205, 1979.

[95] G. Moretti and G. Bleich. A time-dependent computational method for blunt-body flows. *AIAA J.*, 4:2136–2141, 1966.

[96] D. Mumford. Bayesian rationale for energy functionals. In *Geometry-Driven Diffusion in Computer Vision*, pages 141–153. 1994.

[97] D. Mumford. *Elastica and Computer Vision.* Springer, New York, 1994.

[98] D. Mumford. *Geometry-Driven Diffusion in Computer Vision.* Kluwer Academic Publishers, Norwell, MA, USA, 1994.

[99] H. H. Nagel. Constraints for the estimation of displacement vector fields from image sequences. In *Int. Joint Conf. on Artificial Intelligence*, pages 945–951, 1983.

[100] H. H. Nagel. On the estimation of optical flow: Relations between different approaches and some new results. *Artif. Intell.*, 33:299–324, 1987.

[101] H. H. Nagel. Extending the 'oriented smoothness constraint' into the temporal domain and the estimation of derivatives of optical flow. In *Proceedings of the first european conference on computer vision*, pages 139–148. Springer, 1990.

[102] H. H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(5):565–593, 1986.

[103] S. Negahdaripour and C.-H. Yu. A generalized brightness change model for computing optical flow. *Proceedings International Conference on Computer Vision Germany*, pages 2–11, May 1993.

[104] O. Nestares and D.J. Fleet. Detection and tracking of motion boundaries. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 358–365, 2001.

[105] M. Nitzberg, D. Mumford, and T. Shiota. *Filtering, Segmentation, and Depth.* Springer, Secaucus, NJ, USA, 1993.

[106] W. A. Oldfather, C. A. Ellis, and D. M. Brown. Leonhard Euler's elastic curves. *Isis*, 20(1):72–160, 1933.

[107] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *Int. J. Comput. Vision*, 67(2):141–158, 2006.

[108] K. A. Patwardhan, G. Sapiro, and M. Bertalmío. Video inpainting of occluding and occluded objects. The 2005 IEEE International Conference on Image Processing, Genova.

[109] K.A. Patwardhan, G. Sapiro, and M. Bertalmío. Video inpainting under constrained camera motion. *IP*, 16(2):545–553, 2007.

[110] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. In *Proc. IEEE Comput. Soc. Workshop Computer Vision (Miami, FL)*, pages 16–27. IEEE Computer Society Press, Washington, 1987.

[111] M. Raffel, C.E. Willert, and J. Kompenhans. *Particle Image Velocimetry. A Practical Guide.* Springer, 1999.

[112] W. Ritz. Über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik. *Z. Ang. Math. Mech.*, 135(1):1–61, 1908.

[113] L. I. Rudin and S. Osher. Total variation based image restoration with free local constraints. In *ICIP*, volume 1, pages 31–35, 1994.

[114] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. In *Proceedings of the eleventh annual international conference of the center for nonlinear studies on experimental mathematics: Computational issues in nonlinear science*, pages 259–268, Amsterdam, The Netherlands, 1992. Elsevier North-Holland, Inc.

[115] P. Ruhnau. *Variational Fluid Motion Estimation with Physical Priors.* PhD thesis, Department of Mathematics and Computer Science, University of Mannheim, 2007.

[116] P. Ruhnau and C. Schnörr. Optical Stokes flow: An imaging-based control approach. *Experiments in Fluids*, 42:61–78, 2007.

[117] P. Ruhnau, A. Stahl, and C. Schnörr. On-line variational estimation of dynamical fluid flows with physics-based spatio-temporal regularization. Springer, Proc. 28th German Pattern Recognition Symposium (DAGM '06). LNCS 4174, 2006.

[118] P. Ruhnau, A. Stahl, and C. Schnörr. Variational estimation of experimental fluid flows with physics-based spatio-temporal regularization. *Measurement Science and Technology*, 18:755–763, 2007.

[119] 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, 1991.

[120] C. Schnörr. Segmentation of visual motion by minimizing convex non-quadratic functionals. In *12th Int. Conf. on Pattern Recognition, Jerusalem, Israel*, volume A, pages 661–663. IEEE Computer Society Press, 1994.

[121] C. Schnörr. Variational methods for adaptive image smoothing and segmentation. *Handbook on Computer Vision and Applications: Signal Processing and Pattern Recognition*, 2:451–484, 1999.

[122] C. Schnörr and J. Weickert. Variational image motion computation: Theoretical framework, problems and perspectives. Springer, 2000.

[123] T. K. Shih, N. C. Tang, W. Yeh, T. Chen, and W. Lee. Video inpainting and implant via diversified temporal continuations. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 133–136, New York, NY, USA, 2006. ACM.

[124] T. Shiratori, Y. Matsushita, S. Bing, and K. X. Tang. Video completion by motion field transfer. In *Proceedings of CVPR 2006*, pages 411–418. IEEE, 2006.

[125] H. Spies, T. Dierig, and C. Garbe. Local models for dynamic processes in image sequences. In *Workshop Dynamic Perception*, pages 59–64, Bochum, Germany, 2002.

[126] A. Stahl, P. Ruhnau, and C. Schnörr. *A Distributed Parameter Approach to Dynamic Image Motion.* International Workshop on The Representation and Use of Prior Knowledge in Vision. ECCV Workshop, 2006.

[127] G. Stockman and L. G. Shapiro. *Computer Vision.* Prentice Hall PTR, 2001.

[128] J. T. Stuart. The Lagrangian picture of fluid motion. *Philosophical Transactions: Physical Sciences and Engineering*, 333(1631):263–271, 1990.

[129] X. Tai, S. Osher, and R. Holm. Image inpainting using a TV-Stokes equation. Technical report, UCLA, CAM-report-06-01, January 2006.

[130] J.W. Thomas. Numerical partial differential equations: Conservation laws and elliptic equations. volume 33. Springer, 1999.

[131] J.W. Thomas. Numerical partial differential equations: Finite difference methods. volume 22. Springer, 1999.

[132] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems.* VH. Winston & Sons, John Wiley & Sons, Washington D.C., 1977.

[133] E. F. Toro. Defects of conservative approaches and adaptive primitive-conservative schemes for computing solutions to hyperbolic conservation laws. Technical report mmu 9401, Department of Mathematics and Physics, Manchester Metropolitan University, UK, 1994.

[134] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics.* Springer, 1999.

[135] D. Tschumperlé. Curvature-preserving regularization of multi-valued images using PDE's. In *ECCV*, volume 2, pages 295–307, 2006.

[136] A. Tveito and R. Winther. *Introduction to Partial Differential Equations: A Computational Approach*, volume 29 of *Texts in Applied Mathematics.* 1998.

[137] B. Van Leer. Towards the ultimate conservative difference scheme I. The quest of monotonicity. *Lecture Notes Phys*, 18:163–168, 1973.

[138] L. A. Vese and S. J. Osher. Image denoising and decomposition with total variation minimization and oscillatory functions. *J. Math. Imaging Vis.*, 20(1-2):7–18, 2004.

[139] S. Walden. *The Ravished Image.* St. Martins's Press, New York, 1985.

[140] L. Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH '00: Proceedings of the 27th annual conference on computer graphics and interactive techniques*, pages 479–488, New York, USA, 2000. ACM Press, Addison-Wesley Publishing Co.

[141] J. Weickert. *Anisotropic Diffusion in Image Processing.* Teubner-Verlag, 1998.

[142] 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, 2001.

[143] J. Weickert and C. Schnörr. Variational optic flow computation with a spatio-temporal smoothness constraint. *J. Math. Imaging and Vision*, 14(3):245–255, 2001.

[144] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *CVPR*, volume 1, pages 120–127, 2004.

[145] G. B. Whitham. *Linear and Nonlinear Waves.* Pure & Applied Mathematics, 1974.

[146] J. Yuan, P. Ruhnau, E. Mémin, and C. Schnörr. Discrete orthogonal decomposition and variational fluid flow estimation. In *Scale-Space 2005*, volume 3459, pages 267–278. Springer, 2005.