Variationelle Optische Flußberechnung

Präzise Modellierung und effiziente Numerik

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

vorgelegt von

Andrés Bruhn

Saarbrücken 2006

Gutachter

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

Prof. Dr. Christoph Schnörr (2. Gutachter)
Universität Mannheim

Prof. Dr. Hans–Hellmut Nagel (3. Gutachter)
Universität Karlsruhe

Zusammenfassung

Die Bestimmung des Verschiebungsvektorfeldes zwischen zwei aufeinander folgenden Bildern einer Bildfolge – des sogenannten optischen Flusses – ist eines der zentralen Probleme des Maschinensehens. Aufgrund ihrer Eigenschaft verschiedene Annahmen mathematisch korrekt im Rahmen eines gemeinsamen Minimierungsproblems formulieren zu können, zählen Variationsansätze zu den qualitativ besten und den am besten verstandenen Verfahren für diese Aufgabe überhaupt. Sie können so modelliert werden, dass sie Bewegungsgrenzen respektieren, große Verschiebungen korrekt behandeln, sich robust gegenüber Beleuchtungswechseln erweisen oder gute Ergebnisse bei Rauschen oder Verdeckungen erzielen. Trotzdem finden sie in der Praxis nicht sehr oft Verwendung, da sie das Lösen großer linearer oder nichtlinearer Gleichungssysteme erfordern. Insbesondere gelten sie deshalb als zu langsam für Aufgaben, die auf eine Berechnung in Echtzeit angewiesen sind.

In dieser Arbeit werden zwei wichtige Beiträge auf dem Gebiet der variationellen optischen Flussberechnung geleistet: Zum einen wird ein *systematischer Rahmen* zur Entwicklung genauer Variationsansätze vorgestellt. Dieser erlaubt den Entwurf der *zur Zeit präzisesten* Schätzverfahren für den optischen Fluss in der gesamten Literatur. Zum anderen wird ein *allgemeiner Ansatz* zur Konstruktion hochperformanter Mehrgitterverfahren präsentiert, der ein schnelles und effizientes Lösen der auftretenden linearen und nichtlinearen Gleichungssysteme sicher stellt. Dieser Ansatz ist nicht nur in seiner Effizienz üblicherweise verwendeten numerischen Verfahren um bis zu *vier Größenordnungen* überlegen, er ermöglicht zudem die erste *Echtzeitberechnung* von variationellem optischen Fluss überhaupt.

Der erste Teil dieser Arbeit befasst sich mit variationellen optischen Flußverfahren zur Schätzung kleiner Verschiebungen. In diesem Rahmen wird die systematische Entwicklung konvexer Ansätze untersucht. Neben der Analyse einer Vielzahl von bewährten Modellierungskonzepten, werden hierbei auch mehrere eigene Ideen vorgestellt, die die Qualität der Schätzung im Bezug auf Rauschen, Ausreißer und wechselnde Beleuchtung verbessern. Darüber hinaus wird eine kompakte Notation für variationelle optische Flussverfahren vorgestellt, die auf Bewegungs- und Diffusionstensoren basiert. Diese Notation erweist sich in zweierlei Hinsicht als nützlich: Zum einen ermöglicht sie den systemtatischen Entwurf neuer Verfahren. Dies wird unter anderem am Beispiel zweier neu entwickelter Modelle gezeigt. Zum anderen bildet sie die Basis eines allgemeinen numerischen Ansatzes zur Entwicklung hocheffizienter bidirektionaler Mehrgitterverfahren. Dieser beruht auf einem direkten Transfer der Tensoren auf gröbere Gitter, anstatt wie sonst üblich eine Unterabtastung der Originalbildfolge durchzuführen. Die daraus resultierende Grobgitterapproximation ist damit nicht nur effizienter, sondern auch numerisch genauer als die meisten bisher verwendeten Strategien. Ausführliche qualitative und quantitative Experimente anhand sieben verschiedener Modellprototypen und sechs numerischer Verfahren zeigen sowohl die Genauigkeit der Modelle als auch die Effizienz der numerischen Löser.

Im zweiten Teil dieser Arbeit werden die vorherigen Beiträge auf den Fall großer Verschiebungen verallgemeinert. Dazu wird zunächst ein achter Prototyp eingeführt, der auf einem nichtkonvexen Ansatz basiert. Um das zugehörige Minimierungsproblem erfolgre-

ich zu lösen, wird eine inkrementelle Fixpunktiteration vorgestellt, die durch eine sukzessive Verfeinerung des Problems bestimmte lokale Minima umgehen kann. In diesem Zusammenhang gelingt es auch, eine theoretische Rechtfertigung für die weit verbreitete Klasse der Bildvorverzerrungsverfahren (Warping-Verfahren) herzuleiten, deren Einsatz bisher ausschließlich algorithmisch motiviert war. Darüber hinaus wird ein Equivalent des Bewegungstensors für große Verschiebungen eingeführt, welches sowohl die Erweiterung der kompakten Notation als auch die Erweiterung des allgemeinen Ansatzes zur Entwicklung hocheffizienter Mehrgitterverfahren ermöglicht. Die erfolgreiche Verallgemeinerung der obigen Konzepten zeigt sich in den zugehörigen qualitativen und quantitativen Experimenten. Dort werden ähnliche Geschwindigkeitszugewinne erreicht wie im Falle kleiner Verschiebungen, jedoch noch genauere Flussschätzungen.

Variational Optic Flow Computation

Accurate Modelling and Efficient Numerics

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

by

Andrés Bruhn

Saarbrücken, Germany
May 4th, 2006

Abstract

The recovery of the displacement field between two consecutive frames of an image sequence – the so-called optic flow – is one of the central problems in computer vision. Allowing a mathematical sound integration of different concepts into a single minimisation framework, variational methods belong to the best performing and best understood techniques for solving this task. They can be designed in such a way that they preserve motion boundaries, treat large displacements correctly, are robust with respect to illumination changes or perform favourably in the presence of noise and occlusions. However, they are hardly used in practical applications, since they require to solve large linear or nonlinear systems of equations. In particular, they are considered to be too slow for those tasks where real-time performance is needed.

In this thesis we make two important contributions to the field of variational optic flow computation: Firstly, we provide a *systematic toolkit* for the design of accurate variational methods. Thereby, we demonstrate that this toolkit allows the construction of the *currently most precise* optic flow approaches in the literature. Secondly, we present a *multigrid framework* for the efficient solution of the resulting linear and nonlinear systems of equations. This framework does not only outperform frequently used numerical schemes by up to *four orders of magnitude*, it even allows the first *real-time* computation of variational optic flow ever.

The first part of this thesis is dedicated to variational optic flow methods for small displacements. Thereby we investigate the systematic design of convex approaches by discussing a variety of established models. In this context, we also introduce several new ideas that improve the quality of the estimation with respect to noise, outliers and varying illumination. Moreover, we present a compact notation for variational optic flow methods that is based on motion and diffusion tensors. This notation proves to be useful in two respects: Firstly, it allows the systematic construction of novel approaches. This is demonstrated by the example of two advanced optic flow techniques. Secondly, it forms the basis of our numerical framework for the design of efficient bidirectional multigrid methods. By resampling these tensors instead of the original image sequence, we present a novel way to create a suitable coarse grid representation that is both computationally efficient and accurate at the same time. Extensive qualitative and quantitative benchmarks for seven different models and six different numerical prototypes show the accuracy and the efficiency of the proposed multigrid implementations.

The second part of this thesis extends our previous contributions to the case of large displacements. To this end, we introduce an eighth prototype that is based on a nonconvex approach. In order to solve the resulting minimisation problem we derive an incremental coarse-to-fine fixed point iteration that allows to avoid local minima. In this context, we also succeed to provide a theoretical justification for the well-known warping technique that has been motivated only on an algorithmic basis so far. Moreover, we define a new motion tensor equivalent for the case of large displacements. This allows us to extend our compact notation and therewith our highly efficient multigrid framework that we developed before. Quality and efficiency benchmarks show the success of this generalisation: While the speedups are similar than in the case of small displacements, the accuracy is even higher.

To Natalia

Acknowledgements

Contents

1

Introduction

"The reasonable man adapts himself to the world; the unreasonable one persists in trying to adapt the world to himself. Therefore all progress depends on the unreasonable man."

- George Bernard Shaw

1.1 Motivation

The estimation of motion information from image sequences is one of the key problems in computer vision. Typically, one is thereby interested in determining the direction and the velocity of moving objects in the scene as well as the motion of the video camera itself – the so-called *ego-motion*. A first step towards obtaining this information and the focus of this thesis is the computation of the *relative motion* of the scene. Hereby, the expression "relative" refers to the fact that we compute the motion relative to the possibly moving camera(s) and not to a static fixed point in the scene. Moreover, we restrict ourselves to *monocular* image sequences – sequences that have been acquired by a single camera only. Projective geometry tells us that in this case the depth of the scene cannot be determined uniquely. As a consequence, the extraction of motion information comes down to computing only a *projection of the actual motion* onto the image plane. This 2-D displacement field that describes the apparent motion of the scene is the so-called *optic flow*. Figure 1.1 illustrates the resulting correspondence problem: Given two consecutive frames of an image sequence, one is interested in determining that displacement vector field that maps all points of the first frame onto their new location in the second frame.
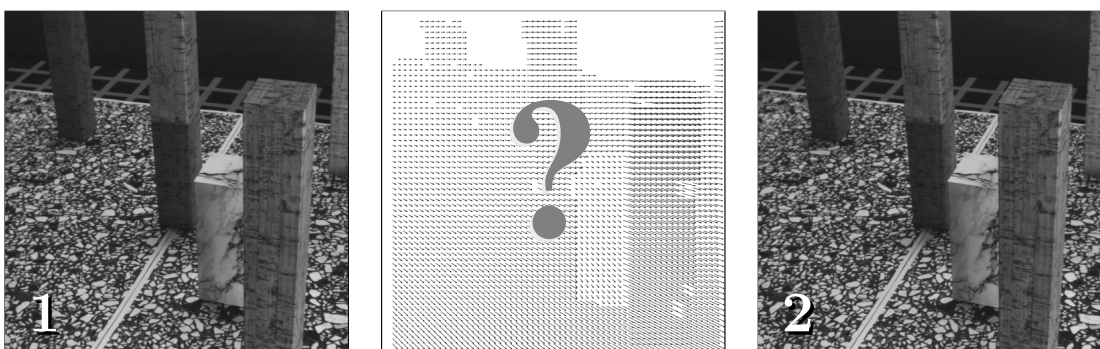
Fig. 1.1: Example for the optic flow problem. **Left and Right:** Two consecutive images of the *Marble* sequence by Otte and Nagel. **Centre:** Wanted displacement field (optic flow). How can we compute this displacement field in a fast and accurate manner?

Fig. 1.2: Possible fields of application. **Left:** Tracking of cars in a traffic scene. **Centre:** Navigation of a car through a street. **Right:** Computed optic flow field. The colour indicates the direction of the relative motion.

Fields of Application. Although the optic flow is only the projection of the true motion of the scene, it proves to be useful for a variety of different tasks (see Figure 1.2). For example, it allows to distinguish stationary from moving objects and thus to detect and avoid obstacles [Enk91]. This makes it particularly useful for tasks where vehicles have to be guided safely through an unknown environment. In this context one can either think of *robot navigation* where one is interested in a fully autonomous behaviour [DK02] or the design of *drivers assistance systems* where support is only required in certain situations [LMB02, Hei02].

Moreover, the estimated motion allows to *track objects* on their way through the scene, to keep them focused and to follow them if desired [KN90, KDTN92, NH98, LMB02]. This interaction with the environment goes far beyond a pure avoidance of obstacles. In combination with approaches from machine learning, motion patterns can be trained in such a way that the obtained algorithms even allow for the interpretation of human mimics and gestures [BYJ97]. These techniques serve as an important step towards *human-machine interfaces*.

Another field of application that is more related to image processing than to computer vision is the *compression* of video sequences. The basic idea is thereby to decompose a sequence of images into a small set of key frames and encode the differences to the remaining frames as flow fields [MKW97, HP01]. Extending this idea to an even more compact representation based on object shapes and single displacement vectors describing their motion, one obtains the specification of the current MPEG-7 compression standard [Mar].

Finally, computing the optic flow is also directly related to other important correspondence problems in computer vision such as *stereo reconstruction* [ADSW02, SBW05] and *image registration* [WPFS05]. While in the case of stereo reconstruction one is interested in finding the mapping between the left and the right frame of a stereo pair, image registration requires to compute the displacement field between two images that may have even been obtained by two different types of image acquisition methods, e.g. computer tomography (CT) and magneto resonance imaging (MRI).

This direct relation to other correspondence problems and the previously discussed practical applications are only a few examples that demonstrate the usefulness of optic flow. However, they clearly show why in the last two decades so much research has been carried out with respect to the development of accurate models and fast numerical schemes for its estimation.

1.2 Overview

The goal of the present work is to contribute in *two ways* to the field of optic flow computation. On one hand, we are interested in improving the *qualitative performance* of current optic flow techniques. To this end, we focus on the class of optic flow approaches that gives the currently best results in the literature – so-called *variational methods*. By discussing existing and novel concepts for designing such methods and by introducing a compact notation that is based on different tensor formulations, we provide a *systematic toolkit* for the construction of novel highly accurate optic flow techniques. On the other hand, we are also interested in a *fast computation* of the results. Therefore, we also present a *general numerical framework* for these methods that is based on one of the most efficient computational schemes – so-called *bidirectional multigrid methods*. This allows us to make variational techniques more appealing for time-critical tasks that are currently dominated by fast but poorly performing optic flow algorithms. By addressing both aspects the *modelling* and the *numerics* of variational methods, this thesis shall provide a common basis for the design of fast and accurate optic flow techniques.

In order to specify our contributions in detail, let us give a short introduction to variational optic flow computation and on the use of multigrid methods in the context of motion estimation. Moreover, we will discuss relevant work that is related to both fields of research.

1.2.1 Variational Optic Flow Computation

Since the prototypical approach of Horn and Schunck [HS81] in 1981, variational methods are among the best performing and best understood techniques for computing the optic flow. Such methods determine the desired displacement field as the minimiser of a suitable energy functional, where deviations from model assumptions are penalised. In general, this energy functional consists of two terms: a *data term* that imposes temporal constancy on certain image features, e.g. on the grey value of objects, and a *smoothness term* that regularises the often non-unique (local) solution of the data term by an additional smoothness constraint. While the data term represents the assumption that certain (characteristic) image features do not change over time and thus allow for a retrieval of corresponding objects in subsequent frames, the smoothness term stands for the assumption that neighbouring points most probably belong to the same object and thus undergo a similar type of motion.

How Do Variational Methods Work? Let us demonstrate this rather general strategy by a concrete example: the classical variational method of Horn and Schunck [HS81]. This method is based on the two most frequently used assumptions in the optic flow literature: the *brightness constancy assumption* which assumes that the grey value of objects remains constant over time and *homogeneous regularisation* which assumes that the resulting flow field is smooth everywhere. In order to formulate these two assumptions mathematically, let us consider a scalar-valued image sequence $f(x_1, x_2, x_3)$, where $(x_1, x_2)^\top$ is the location within a rectangular image domain $\Omega_2 \subset \mathbb{R}^2$ and $x_3 \in [0, T]$ denotes time. Furthermore, let us assume that the temporal displacement rate between two consecutive frames
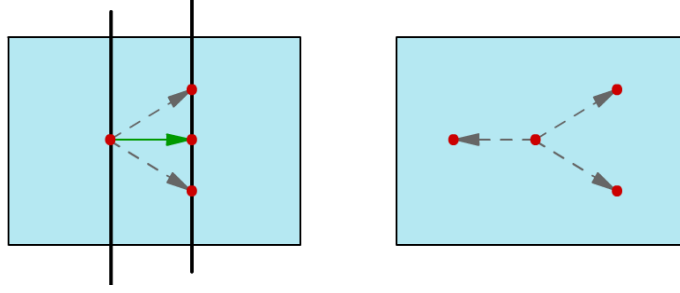
Fig. 1.3: **(a) Left:** Aperture problem. Only the component of the flow orthogonal to $\nabla_2 f$ is computable (green arrow). **(b) Right:** No information available. Correspondences may lie everywhere.

of the image sequence is $\Delta x_3 = 1$. Then, the assumption of a constant image brightness can be formulated as

$$f(x_1+u_1,\ x_2+u_2,\ x_3+1) - f(x_1, x_2, x_3) = 0\,,$$

where $\mathbf{u}(x_1, x_2, x_3) = (u_1(x_1, x_2, x_3), u_2(x_1, x_2, x_3), 1)^\top$ is the desired optic flow field. If we furthermore assume that the displacements are small and that the image sequence f varies smoothly in space and time, we can approximate this equation reasonably well by performing a first-order Taylor expansion. Then, the following linearised *optic flow constraint* (OFC) is obtained

$$f_{x_1} u_1 + f_{x_2} u_2 + f_{x_3} = 0\,,$$

where subscripts denote partial derivatives. Evidently, this single equation is not sufficient to determine the two unknowns u_1 and u_2 uniquely. In the literature this problem is referred to as the *aperture problem*: For non-vanishing image gradients, it is only possible to determine the flow component parallel to $\nabla_2 f := (f_{x_1}, f_{x_2})^\top$, i.e. normal to image edges; see Figure 1.3 (a). For $\nabla_2 f \neq 0$ this so-called *normal flow* is given by

$$\mathbf{u}_n = (-\frac{f_{x_3}}{|\nabla_2 f|} \frac{\nabla_2 f^\top}{|\nabla_2 f|}, 1)^\top. \qquad (1.1)$$

At locations where $\nabla_2 f = 0$ the situation even becomes worse: Since there is no image information available, not even this normal flow can be calculated; see Figure 1.3 (b).

As indicated before, the method of Horn and Schunck overcomes this locally non-unique solution by assuming the resulting flow field to be globally smooth. This can be formulated mathematically by penalising large spatial flow gradients $\nabla_2 u_1$ and $\nabla_2 u_2$. Combining both the constancy and smoothness assumption in a single variational framework and squaring both constraints in order to penalise negative and positive deviations in the same way, the following energy functional is obtained:

$$E_{\mathrm{HS}}(\mathbf{u}) = \int_{\Omega_2} \Big(\underbrace{(f_{x_1} u_1 + f_{x_2} u_2 + f_{x_3})^2}_{\text{data term}} + \alpha \underbrace{\big(|\nabla_2 u_1|^2 + |\nabla_2 u_2|^2\big)}_{\text{smoothness term}} \Big)\, d\mathbf{x}, \qquad (1.2)$$

where $\mathbf{x} := (x_1, x_2)^\top$. The degree of smoothness of the solution is thereby steered by the positive weight α that serves as a *regularisation parameter*: The larger the value of α, the smoother becomes the computed flow field.

Tab. 1.1: Definition of the integration domain Ω, the integration variable $\mathbf{x}$ and the gradient operator ∇ for spatial and spatiotemporal energy functionals.

	spatial case (2 frames)	spatiotemporal case (> 2 frames)
Ω	$\Omega := \Omega_2 \subset \mathbb{R}^2$	$\Omega := \Omega_3 \subset \mathbb{R}^2 \times [0, T]$
$\mathbf{x}$	$\mathbf{x} := (x_1, x_2)^\top$	$\mathbf{x} := (x_1, x_2, x_3)^\top$
∇	$\nabla := \nabla_2 := (\partial_{x_1}, \partial_{x_2})^\top$	$\nabla := \nabla_3 := (\partial_{x_1}, \partial_{x_2}, \partial_{x_3})^\top$

A General Energy Functional. After we have explained the basic strategy of variational techniques by the example of the method of Horn and Schunck, let us now clarify in which types of energy functionals we are interested in this thesis. To this end, let us define a more general energy functional. If we denote by $D^k f$ the set of all partial (spatial and temporal) derivatives of f of order k, then, this energy functional can be formulated as

$$E(\mathbf{u}) = \int_\Omega \big(\underbrace{M(D^k f, \mathbf{u})}_{\text{data term}} + \alpha \underbrace{S(\nabla f, \nabla \mathbf{u})}_{\text{smoothness term}} \big) \, d\mathbf{x}, \qquad (1.3)$$

where $M(D^k f, u)$ denotes a data term that represents one or more constancy assumptions on $D^k f$ that are either given in their original nonlinear or their linearised form, while $S(\nabla f, \nabla \mathbf{u})$ stands for a smoothness term that assumes the flow field $\mathbf{u}$ to be smooth or piecewise smooth. In order to preserve discontinuities, this term can be adapted to the image gradient ∇f or to the flow gradient $\nabla \mathbf{u}$ itself. Moreover, it can be extended to the spatiotemporal domain, which requires a redefinition of the integration domain Ω, of the integration variable $\mathbf{x}$ and of the gradient operator ∇ (see Table 1.1). As one can easily verify, the approach of Horn and Schunck in (1.2) is one particular instance of this general energy functional. However, in this thesis we are interested in much more elaborated approaches that make use of the full potential of our general formulation.

▶ Our first main goal in this thesis is to provide a general toolkit for the systematic construction of variational optic flow methods based on our general energy functional in (1.3). In order to achieve it, we will discuss various established concepts for the design of both the data and the smoothness term. Apart from investigating existing strategies, we will also present several novel ideas that address typical problems in motion estimation such as noise, varying illumination and outliers.

Convexity vs. Nonconvexity. In general, it is desirable to design the energy functional in (1.3) in such a way that it is *strictly convex*. In this case the minimisation is rather simple, since a *unique minimiser* exists that can be found by any globally convergent algorithm. However, as in the case of the method of Horn and Schunck, the construction of such an energy functional requires the *linearisation* of all constancy assumptions. As long as we restrict ourselves to image sequences with small displacements, this does not influence the estimation quality, since the linearised assumptions are valid approximations to the original nonlinear ones. In contrary, the convex design even offers additional advantages such as *well-posedness* results and the *continuous dependency of the solution* on both

the input data and the model parameters [ADK99, AK02, WS01a, HSSW02]. However, if we are interested in estimating displacements that are significantly larger than a few pixels per frame, this method suffers from their linearised assumptions and performs relatively poor. In this case one has to replace this assumptions and use their original nonlinear counterparts instead. This in turn results in an energy functional that is *nonconvex*. Such a functional requires the use of much more sophisticated optimisation strategies, since algorithms that can be applied in the convex case get easily trapped in a local minimum very close to the initialisation.

▶ With respect to the design of variational methods we will discuss both the convex and the nonconvex case. In the latter case we will also present a suitable minimisation framework that allows to find suitable local minima.

Why Variational Methods? Although we have already explained the basic idea of variational methods and discussed the advantages of convex functionals, we only briefly motivated so far why we have chosen exactly this class of optic flow techniques. This is in particular important with respect to the fact that there are also numerous other strategies in the literature to solve the underlying correspondence problem: *Local differential methods* that are based on the same constancy assumptions as variational techniques but minimise local energy-like expressions [LK81, Luc84, BGW91, Far01], *feature-based techniques* that seek correspondences for sparse but characteristic image features such as edges or corners [BB84, WWB88, WAB06], *area-based approaches* that rely on matching complete image patches by aggregating local information [Ana89, Sin90], and *phase-based approaches* that make use of velocity-tuned filters in the Fourier domain [FJ90, Fel05]. Let us now discuss the four main advantages that variational methods offer when compared to the aforementioned classes of optic flow techniques:

+ *Transparent Modelling.* Variational methods allow for a transparent modelling by construction: All assumptions on the image data and the solution are explicitly formulated in the underlying energy functional. There are no intermediate or postprocessing steps that question the consistency of the whole approach. Moreover, the use of a joint minimisation framework allows thereby a mathematical sound integration of all desired assumptions.

+ *Invariance under Rotations.* If designed in the continuous setting, these energy functionals can formulated in such a way that they are invariant under rotations. Results from numerical analysis show that consistent discretisations approximate this invariance under rotations arbitrarily well if the sampling is sufficiently fine.

+ *Dense Flow Fields.* Variational optic flow methods are *global* methods: If there is not sufficient local information, the data term $M(D^k f, \mathbf{u})$ is so small that it is dominated by the smoothness term $\alpha S(\nabla f, \nabla \mathbf{u})$ which fills in information from more reliable surrounding locations. Thus, in contrast to local methods, the *filling-in effect* of global variational approaches always yields dense flow fields such that no subsequent interpolation steps become necessary. Everything is automatically accomplished within a single variational framework.

+ *Better Qualitative Performance.* As shown in different performance evaluations [BFB94, GMN^{+}98] and in the recent literature on optic flow [MP98a, BWS05, PBB^{+}06, AK05a], variational methods are those techniques that currently offer the *highest precision* in terms of error measures [BFB94].

These advantages make the superiority of variational optic flow methods explicit. However, there is a also a *price to pay* when using such techniques: The regulariser that yields the desired filling-in-effect also leads to a coupling of neighbouring pixels in the optimisation problem. This in turn requires a joint optimisation of the solution for all pixels by means of solving one or more *large linear or nonlinear systems of equations*. Often standard iterative numerical schemes are applied for this purpose such as the *Jacobi* or the *Gauß-Seidel* method [OR00, You71, Mei99]: While they are easy to implement, they require thousands of iterations to reach a desired accuracy.

▶ Our second main goal in this thesis is the development of a numerical framework for the efficient solution of these linear and nonlinear systems of equations. Thereby, we want to show that high quality optic flow computation is possible in real-time. In order to achieve it, we will discuss a variety of numerical schemes. In particular, we will focus on multigrid methods, since they are among the fastest techniques for solving this task.

1.2.2   Multigrid Methods for Variational Optic Flow Computation

There are basically three different types of multigrid methods that are used in the context of variational optic flow methods: *unidirectional multigrid methods*, *unidirectional warping techniques* and *bidirectional multigrid methods*. They are either applied to speed up the computation or to improve the quality of the results when dealing with nonconvex optimisation problems. Let us now briefly discuss all three types of techniques.

Unidirectional Multigrid Methods. The most popular class of multigrid methods are so-called *unidirectional* schemes [Ana89, BAK91, LKW94, BD96]. These coarse-to-fine strategies start with a coarse version of the original problem and refine it during the solution process. Thereby, coarse grid solutions serve as initialisations on finer grids. Unidirectional multigrid schemes offer two advantages: One one hand – in the case of *convex* energy functionals – they allow to speed up the *computation* significantly; see e.g. unidirectional schemes in [BWF^{+}05]. This is not surprising, since coarse grid results are in general good approximations to fine grid solutions. However, one cannot expect that the numerical algorithm converges to a better solutions, since a strictly convex energy functional has only a single minimiser. On the other hand – in the case of *nonconvex* energy functionals – they also allow to improve the *quality* of the results significantly [BA96, MP98b, AWS00, BWS05, PBB^{+}06]. This is a direct consequence of the fact that certain undesired local minima disappear at sufficiently coarse resolutions and thus can be avoided by a coarse-to-fine framework.

Unidirectional Warping Techniques. In the nonconvex case – in particular in the context of large displacements – unidirectional multigrid schemes are often combined with so-called *warping steps* [BAHH92, BA96, MP98a]. Thereby the original problem is

compensated by the already computed motion from all coarser levels before the resolution is refined. What remains to be solved at each resolution level is the motion increment for the difference problem [BA96, MP98b, AWS00]. Such a proceedings allows to keep the displacements at each resolution level small, so that linearised constancy assumptions remain reasonable approximations. As is to be expected the obtained results for large displacements are much better than to approaches with linearised constancy assumptions that are used only on one resolution level – such as the ones in [HS81, NE86, AELS99, WS01a, WS01b].

Bidirectional Multigrid Methods. From a numerical viewpoint, unidirectional multigrid schemes are not the end of the road. Very promising – in particular in the context of variational methods – are so-called *bidirectional* multigrid methods [Bra77, Hac85, Wes92, BHM00, TOS01]. These techniques that create a sophisticated hierarchy of equation systems with excellent error reduction properties belong to the fastest numerical schemes for solving linear or nonlinear systems of equations. In contrast to unidirectional schemes they revisit coarser resolution levels in order to obtain useful correction steps. Thus, they are able to overcome the typical limitation of basic iterative solvers that is also present in unidirectional multigrid schemes: *the weak attenuation of low error frequencies* [Bra77, Hac85]. Moreover, they can benefit from all advantages of unidirectional multigrid schemes, since they may use the same coarse-to-fine initialisation strategy on top. Then, these bidirectional multigrid schemes are referred to as *full* multigrid methods [BHM00].

▶ With respect to the design of multigrid methods we will analyse the construction of all three types of techniques. Moreover, we will seek to combine the different strategies in order to obtain a numerical scheme that allows for both an accurate estimation and an efficient performance.

1.3   Related Work

After we have clarified the main goals of this thesis and indicated how we will reach them, let us now discuss some relevant work that is related to our two most important contributions: the modelling of variational optic flow methods and the usage of bidirectional multigrid methods in motion estimation.

1.3.1   Modelling of Variational Methods

The modelling of variational methods for computing the optic flow goes back to the prototypical approach of Horn and Schunck in 1981 [HS81]. Since then, a lot of research has been carried out to improve the performance of such techniques. Due to the vast amount of literature, concepts for the design of data and smoothness terms are discussed separately. Moreover, we will restrict ourselves only to the most relevant work. More detailed references are given throughout the Chapters 2 and 5, where the design of variational methods is studied.

Data Term. With respect to the design of data terms, there are two main fields of research in the literature: *Robust data terms* that use non-quadratic penaliser functions to

improve the performance in the presence of outliers in the image data [BA91, BA96, MP98a, HSSW02] and *modified constraints* that allow for a more accurate estimation in different situations such as varying illumination, large displacements and noise [Nag83a, NE86, Sch93, Sch94a, AWS00, BWS02, BW02].

- *Robust Data Terms.* The design of robust data terms for variational optic flow methods goes back to Black and Anandan [BA91, BA96] who suggested the use of M-estimators from robust statistics [Hub81]. Such functions penalise outliers less severely than the quadratic function originally proposed by Horn and Schunck and thus reduce the influence of corrupted data on the result. Some years later, also Mémin and Pérez [MP98a] investigated the usefulness of M-estimators but solved the resulting nonconvex optimisation problem by an iteratively weighted least square fit instead of using continuous annealing as proposed by Black and Anandan. For *continuous* quasi-convex energy functionals similar non-quadratic growth functions were studied by Hinterberger *et al.* in [HSSW02]. In this context also a regularised variant of the L_1 estimator was investigated. Such an estimator is of particular interest, since it provides both robustness and convexity.

- *Modified Constraints.* Regarding the construction of modified constraints for the data term a variety of ideas has been proposed in the literature. Suitable constraints with respect to a changing image brightness have been suggested by Schnörr [Sch93, Sch94a]. For a more accurate estimation of small displacements Nagel proposed the approximation via a second order Taylor expansion [Nag83b], while Nagel and Enkelmann as well as Alvarez *et al.* proposed to tackle large displacements by constancy assumptions without linearisations [NE86, AWS00]. In order to improve the performance of the data term with respect to noise, Bruhn *et al.* proposed the integration of a local least square fit [BWS02]. This idea was further refined by Brox *et al.* who proposed ro replaced the static averaging of the least square fit by an adaptive one based on nonlinear diffusion [BWS02, BW02].

Smoothness Term. In the case of the smoothness term the literature mainly focuses on two aspects: The *preservation of motion discontinuities* [Nag83a, NE86, SH89, BA91, Coh93, HB93, Nes93, PVPO94, Sch94b, KTB96, AELS99, ADK99, WS01a] and the *integration of temporal information* [Nag90, BA91, WS01b].

- *Discontinuity-Preserving Smoothness Terms.* The first approach to design adaptive regularisers goes back to Nagel in 1983 who proposed an oriented smoothness constraint that inhibits the filling-in-effect across *image* discontinuities [Nag83a, NE86]. Some years later, in the context of *discrete* variational approaches based on Markov random fields (MRF), Shulman and Herve, Heitz and Bouthemy as well as Nesi presented an alternative approach based on the use of robust statistics [SH89, Nes93, HB93]. By penalising outliers in the smoothness less severely than Horn and Schunck, they explicitly allowed for discontinuities in the unknown flow field. Thus, they overcame the typical problem of image-driven regularisers that reduce smoothness in particular in textured regions. Similar non-quadratic regulariser that steer the smoothness in accordance with the evolving *flow field* have also been

suggested by Schnörr and Weickert, however, in the context of *continuous* formulations of variational techniques [Sch94b, WS01a]. Thereby, in particular the work in [WS01a] is of interest, since it proposes a classification of continuous smoothness terms based on their induced diffusion process. Similar smoothness terms that also respect discontinuities of the evolving flow field but are *not invariant under rotations* have been proposed first by Cohen [Coh93] and Deriche *et al.* [DKA95].

- *Spatiotemporal Regularisers.* The first proposal to model spatiotemporal smoothness terms goes back to Murray and Buxton [MB87]. in 1987 they proposed to use a *discrete* optic flow method based on spatiotemporal Markov random fields in order to estimate multiple flow fields simultaneously. Shortly thereafter, Nagel extended his spatially oriented smoothness constraint from [Nag83a] to the temporal domain and thus introduced the first spatiotemporal smoothness term for an *continuous* approach [Nag90]. Another strategy to make use of temporal information was later on suggested by Black [BA91]. Instead of assuming piecewise smoothness in a spatiotemporal data volume, he used previously computed flow estimates as prior knowledge and thus obtained temporally piecewise homogeneous motion fields. This back-in-time adaptation strategy was improved by Weickert and Schnörr who extended their discontinuity-preserving regularisers to the temporal domain [WS99, WS01b]. Thus, smoothness terms were obtained that allow the preservation of both spatial and temporal discontinuities in the unknown flow field and this in forward and backward direction (as Murray and Buxton [MB87] as well as Nagel [Nag90] by a simultaneous computation of multiple flow fields).

▶ The detailed discussion of the preceding four concepts – two for the data and two for the smoothness term – forms the basis of our systematic toolkit for the design of accurate variational optic flow methods in this thesis. Their combination with novel own ideas will thereby allow the construction of the currently most precise optic flow techniques in the literature.

1.3.2   Bidirectional Multigrid Methods for Motion Estimation

In contrast to the modelling of variational optic flow approaches that is a fruitful field of research since decades, only a few works exist in the literature that deal with the construction of bidirectional multigrid methods for variational optic flow techniques.

Linear Bidirectional Multigrid Methods. The first approaches to use liner bidirectional multigrid methods for computing the optic flow go back to Glazer [Gla84], Terzopoulos [Ter86] and Enkelmann [Enk87]. Since also the research of optic flow techniques was a relatively new field of research at that time, the implemented approaches were restricted to simple optic flow techniques: While Glazer and Terzopoulos investigated the efficient implementation of the prototypical method of Horn and Schunck [HS81], Enkelmann focused its multigrid efforts on implementing a modified variant of a technique proposed by Nagel [Nag83a]. But also from a numerical viewpoint, these first multigrid approaches could not convince: Due to the use of an *inappropriate coarse grid approximation strategy* that was based on a resampling of the image data, they offered a relatively *poor performance* or showed even *convergence problems* for an increasing number of multigrid

levels. This was also critisised by Battiti *et al.* [BAK91] who proposed to refrain from bidirectional multigrid methods and to develop adaptive unidirectional multigrid schemes instead.

Several years later, Ghosal and Vaněk [GvV96] revived the research of multigrid techniques for variational optic flow methods. They proposed an *algebraic* bidirectional multigrid method for solving a modified variant of the technique of Nagel and Enkelmann [NE86]. In contrast to the *geometric* multigrid schemes of Glazer, Terzopoulos and Enkelmann that were defined on regular grids, their algebraic method was based on graph structures [BHM00]. Thus, at the expense of additional computational costs for the complex data handling, they were able to create an adaptive and thus very accurate coarse grid representation. As a consequence, the presented results were much better than those reported one decade ago. However, with respect to geometric multigrid methods, not much had changed: Shortly after Ghosal and Vaněk, Zini *et al.* [ZSL97] suggested a multigrid implementation for a 3-D Horn and Schunck variant with an additional incompressibility constraint. Unfortunately, only the model was new. For the construction of the coarse grid equation systems the same inappropriate strategy was used as in the approaches of Glazer, Terzopoulos and Enkelmann.

This stagnation in the development of geometric multigrid methods for variational optic flow techniques ended recently when El Kalmoun and Rüde [ER03] proposed a *Galerkin* approach [BHM00] for the method of Horn and Schunck. Instead of resampling the image sequence and creating the equation systems from scratch, they proposed to tackle the problem directly and resample the equation system itself. While this strategy yielded a very efficient multigrid method in terms of iterations, it introduced another problem: The creation of the coarse grid equation systems via the Galerkin approach was computationally very expensive. Therefore, the same authors also proposed a second technique based on the resampling of the image data. However, this technique showed the weaknesses of all previous geometric approaches: It did not converge if more than three multigrid levels were used.

Nonlinear Bidirectional Multigrid Methods. Apart from the previously discussed attempts to develop multigrid schemes for optic flow techniques that require to solve *linear* systems of equations, there also exists one work in the literature that deals with the solution of *nonlinear* systems of equations related to an optic flow approach. In [BIK02], Borzi *et al.* proposed to solve the control equations of a control-theory-based optic flow formulation by means of a nonlinear bidirectional multigrid method – a so-called *full approximation scheme* (FAS) [Bra77]. However, for "ordinary" formulations of optic flow methods that yield nonlinear equation systems such as the ones in [Sch94b, WS01a, WS01b, HSSW02] that form special cases of our general energy functional in (1.1), no attempts have been made so far to develop suitable bidirectional multigrid schemes.

Multigrid Approaches in Image Processing and Computer Vision. Also for other tasks in image processing and computer vision, bidirectional multigrid methods have been developed. In the context of photometric stereo and image binarisation Kimmel and Yavneh [KY03] developed an algebraic multigrid method, while Chan *et al.* [CCW97] researched geometric multigrid schemes for variational deconvolution with total variation (TV) regularisation. For TV denoising Vogel [Vog95] proposed the use of a linear multigrid

method within a nonlinear fixed-point iteration, while, very recently, Frohn-Schnauf *et al.* [FSHW04] investigated a nonlinear multigrid scheme (FAS) for the same task.

▶ In this thesis we will develop a suitable coarse grid representation for geometric bidirectional multigrid methods that is a good compromise between accuracy and efficiency. This representation forms the basis of our general numerical framework for the construction of efficient multigrid methods for variational optic flow techniques. Thereby both the linear and the nonlinear case will be addressed.

1.4   Organisation and Contributions

Different parts of the work presented in this thesis have been published at conferences [BW02, BWF$^+$03, WBPB03, PBBW03, BBPW04, BWF$^+$05, BW05] or journals [BWS05, BWKS05, PBB$^+$06, WBBP06, BWKS06]. With respect to a better readability the main contributions are split into four chapters – three chapters on variational methods for small displacements and one on variational methods for large displacements.

Optic Flow For Small Displacements – Theory

In Chapter 2 we investigate the modelling of variational optic flow methods for *small* displacements, i.e. the design of variational approaches that are based on linearised constancy assumptions. Thereby we show by a detailed discussion of established concepts for both the data and the smoothness term how such methods can be constructed and how the different strategies influence the estimation quality.

In this context, we also present four novel ideas for the data term: (i) We propose constancy assumptions based on higher order image derivatives in order to address the problem of additive or multiplicative illumination changes. (ii) Moreover, we suggest the use of motion-invariant image features in order to overcome the theoretical limitation of constancy assumptions that contain directional information. Experiments show in both cases the superiority of the resulting data terms if compared to classical approaches based on the image brightness [HS81]. (iii) Furthermore, we propose the combination of several constancy assumptions within in a single data term. This keeps the model flexibly and allows for a more reliable estimation of the optic flow if no prior knowledge on the scene is available. In this context, we also introduce the concept of separate robustification, that employs a robust penaliser function to each of the constancy assumptions separately. (iv) Finally, we also extend the combined-local-global (CLG) framework that was presented in [Bru01]. By supplementing the least squares fit of the original CLG approach with a non-quadratic penaliser, we obtain a data term that is not only robust with respect to noise but also with respect to outliers.

Apart from all these novelties, we also introduce a compact notation for variational optic flow methods that is based on motion and diffusion tensors. We show how such tensors can be derived from the data and the smoothness term, respectively, and why they carry all essential information on the underlying optic flow approach. Moreover, we demonstrate why this notation not only allows for a systematic construction and implementation of novel techniques, but also for a better understanding of existing ones.

Optic Flow For Small Displacements – Prototypes

In Chapter 3 we exploit our framework of Chapter 2 and present seven variational methods for small displacements. These methods that shall serve as prototypes for our qualitative and numerical studies comprise two classes of techniques: five basic approaches that cover the most popular strategies for the smoothness term [WS01a] and two advanced techniques that address two important problems in computer vision: noise and varying illumination. While the basic approaches are adapted almost directly from the literature, we develop the advanced prototypes completely from scratch. By combining different concepts introduced in Chapter 2 we thereby show, how our framework for the design of variational methods can actually be applied.

In an extensive experimental evaluation with various synthetic and real-world scenarios, we investigate the advantages and shortcomings of the proposed prototypes. Thereby in particular the two novel approaches yield excellent results.

A further important contribution besides the design and the evaluation of the different prototypes is our successful approach to establish a *taxonomy* for variational optic flow methods based on the structure of their underlying Euler–Lagrange equations. This allows us to reduce our numerical efforts in Chapter 4 to three main cases: a linear one and two nonlinear ones. For all these case we present a suitable discretisation scheme and discuss the resulting linear and nonlinear systems of equations with respect to their coupling of unknowns.

Optic Flow For Small Displacements – Solvers

In Chapter 4 we discuss various numerical schemes for solving the linear and nonlinear equation systems that we obtained for the three general cases in Chapter 3. For each of these cases we investigate the construction of four different types of solvers: (i) basic non-hierarchical iterative solvers such as the Gauß-Seidel method and its block relaxation variants, (ii) advanced non-hierarchical iterative solvers such as the successive overrelaxation technique (SOR), (iii) unidirectional multigrid methods in form of coarse-to-fine strategies and (iv) bidirectional multigrid methods such as ordinary full multigrid methods and full multigrid methods based on full approximation schemes.

Since they are among the fastest techniques for solving linear or nonlinear systems of equations, our focus lies thereby on the development of the *bidirectional multigrid methods*. In this context, we show (i) which basic solvers are appropriate, (ii) propose a suitable pair of intergrid transfer operators and (iii) derive a coarse grid approximation strategy that is both computationally efficient and accurate at the same time. This in turn allows us not only to derive suitable bidirectional multigrid schemes for all our prototypes, but also to establish a complete multigrid framework for the design of efficient hierarchical solvers that can be applied to all variational models that are constructed in accordance with our systematic toolkit from Chapter 2.

In order to evaluate the efficiency of the different numerical schemes, we perform benchmarks for all seven prototypes with image sequences of size 160×120. Thereby, the proposed bidirectional multigrid methods achieve speedups of two to four order of magnitude compared to standard numerical schemes. This in turn corresponds to frames rates up to 63 frames per second which constitutes the first real-time computation of variational optic flow in the literature.

Optic Flow For Large Displacements – Extensions

In Chapter 5 we extend our framework for the accurate modelling and efficient minimisation of variational optic flow methods to *large displacements*. To this end, we introduce an eighth prototype that is based on constancy assumptions in their original nonlinear form. Since the resulting energy functional is nonconvex, we propose its minimisation by means of an incremental coarse-to-fine fixed point iteration. Such a proceedings postpones the linearisation of the constancy assumptions from the model to the optimisation scheme. We show that this late linearisation does not only improve the estimation quality with respect to large displacements, but that it also allows a theoretical justification of the well-known warping technique that has only been motivated on an algorithmic basis so far.

Not surprisingly, this theoretically sound modelling leads to excellent qualitative results. Various quantitative and qualitative experiments show that the proposed prototype does not only outperform all seven techniques from Chapter 3 in most cases, it even produces the currently most accurate flow fields in the literature.

Apart from these contributions to the design and minimisation of variational methods, we also present an efficient numerical scheme for solving the nonlinear equation system at each incremental fixed point step. Thereby we exploit the fact that each of the systems can be seen as a specific instance of one of the nonlinear cases in Chapter 4. Thus it becomes possible to generalise our framework for the design of efficient multigrid schemes also to the case of large displacements.

This is also reflected in the corresponding efficiency benchmarks: As in the case of small displacements, speedups of two to four order of magnitude are achieved compared to frequently used iterative methods. This corresponds to real-time near frame rates of three high quality optic flow fields per second.

2 Optic Flow For Small Displacements – Modelling

"Even a journey of a thousand miles begins with a small step"
- Chinese saying

This chapter discusses the modelling of variational optic flow methods for *small displacements*. As indicated in the introduction, such methods compute the optic flow field as minimiser of an energy functional with *linearised* constancy assumptions. Although limited in their estimation range, these methods are very interesting from an optimisation point of view; They can be designed in such a way that the resulting energy functionals are *convex*. This in turn means, that only one (*unique*) minimiser exists which can easily be found by any globally convergent algorithm. Moreover, *well-posedness* results can be established that show the continuous dependency of the solution on both the input data and the model parameters [ADK99, AK02, WS01a, HSSW02].

In the first part of this chapter we investigate the design of *data terms* for this particular family of variational methods. Starting from our general formulation of an optic flow functional in (1.3) we analyse different constancy assumptions and their corresponding data terms in detail. Thereby in particular the problem of *varying illumination* is addressed. Furthermore, we show how these data term can be modified in such a way that they become *more robust with respect to noise and outliers*. In this context, different concepts such as local least square regressions, adaptive averaging via nonlinear diffusion and statistically robust error norms are discussed.

In the second part of this chapter we investigate the construction of different *smoothness terms*. To this end, we make use of the *diffusion based taxonomy* in [WS01a] that links optic flow regularisers to their underlying diffusion processes. Thereby we show in detail how the different regularisation strategies can be derived and how they realise the *preservations of discontinuities*. Moreover, we discuss how the corresponding smoothness terms can be extended to the *spatiotemporal domain*, i.e. how the information of more than two frames can be used to improve the quality of the optic flow field.

While discussing different solutions and presenting novel approaches for some of these problems, we perform a variety of experiments at the same time. These experiments shall accompany the reader through our detailed discussion and visualise the impact of all concepts on the overall result in both a *quantitative* and a *qualitative* way.

However, this chapter shall not only provide further insights into the design of variational methods for small displacements. It shall also serve as useful tool-kit for the construction of new approaches. By introducing a modular notation based on motion and diffusion tensors we set up a *general framework* that allows for both a simple development and a straightforward implementation of such techniques. While in this and the next chapter, our framework is limited to the design of methods for small displacements, its actual generality becomes obvious in the Chapters 4 and 5. There, it is extended to the development of variational methods for large displacements as well as to the systematic design of efficient numerical schemes for both types of approaches.

2.1   Presmoothing Step

Before we analyse the different assumptions and concepts for the smoothness and the data term let us briefly discuss a common step in the computation of optic flow: *the presmoothing of the image data*. In general, the original sequence f_0 is thereby replaced by a preprocessed variant f that has been obtained via convolution with a Gaussian kernel K_σ of standard deviation σ:

$$f = K_\sigma * f_0. \tag{2.1}$$

This so-called *presmoothing step* has the purpose of removing noise and other destabilising high frequencies that may otherwise spoil the result. Moreover, it guarantees that $f \in C^\infty$, which is in general useful to establish any well-posedness proof for the presented methods. Since the amount of smoothing in this preprocessing step is directly related to the noise level of the input data, the standard deviation σ of the Gaussian kernel is also referred to as *noise scale* [BWS02, BWS05].

The impact of such a presmoothing step on the input data is illustrated in Figure 2.1. Thereby it becomes obvious that a too large amount of presmoothing may be problematic in two ways: Firstly, it removes useful details that may allow to establish better correspondences and secondly, it leads to the estimation of flow fields that may not approximate the ones for the original image sequence too well.
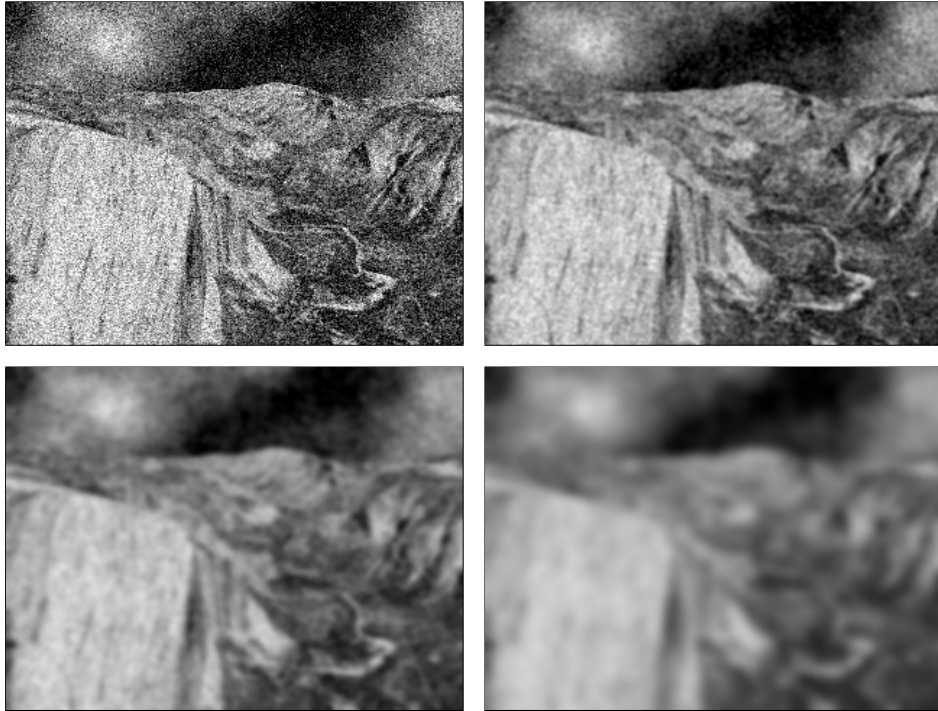
Fig. 2.1:   Presmoothing step. **(a) Top Left:** Frame 8 of the Yosemite sequence *with* clouds with Gaussian noise of standard deviation $\sigma_n = 40$. **(b) Top Right:** Presmoothed variant with $\sigma = 10$. **(c) Bottom Left:** Presmoothed variant with $\sigma = 20$. **(d) Bottom Right:** Presmoothed variant with $\sigma = 40$.

2.2 Data Terms

In the design of data terms for optic flow methods prior knowledge plays an important role. This knowledge includes information on the imaging device (e.g. the quality of the images with respect to noise), on the conditions during the acquisition of the video material (e.g. the occurrence of frequent illumination changes) as well as information on the expected type of motion (e.g. mainly translational motion of cars in traffic sequences). For a specific problem, this information may allow to select a data term that is especially appropriate and thus improves the quality of the estimation significantly. For this reason, the following section gives an overview on data terms that are frequently used in the literature. Moreover, a detailed discussion on their advantages and shortcomings should guide the reader to select an appropriate data term for a specific situation.

2.2.1 Constancy Assumptions - Modelling

The first step of constructing a data term is the choice of one or more suitable constancy assumptions. Such constancy assumptions have to be imposed on certain image features, in order to allow the retrieval of pixels, structures and objects within subsequent frames of the image sequence. Evidently, these image features should be chosen in accordance with the prior knowledge on the scene, since an estimation of correspondences can only be successful if the selected features are invariant under the prevailing illumination conditions and the occurring type of motion. In the following we give a survey on different constancy assumptions that proved to be useful in the context of motion estimation. Thereby we start with the most popular constancy assumption in the literature: the brightness constancy.

2.2.1.1 Constancy Assumption on the Image Brightness

Knowing the lighting conditions to be constant from one frame to the next, a simple but suitable choice for an invariant image sequence feature is the image brightness. To be more precise, one can assume that the grey value of objects does not change over time. Although this classical assumption has already been used in the prototypical approaches of Horn and Schunck [HS81] and Nagel and Enkelmann [NE86] more than two decades ago, it still forms the basis of many accurate optic flow techniques that have recently been developed (see e.g. [MP02] and [BWS05]).

A1 - Constancy of the Brightness. Mathematically, the assumption of a constant image brightness can be expressed by a single constraint. Given two consecutive frames of an image sequence at time x_3 and $x_3 + 1$, respectively, it can be formulated as

$$0 \;=\; f(x_1+u_1,\, x_2+u_2,\, x_3+1) - f(x_1, x_2, x_3). \tag{2.2}$$

Unfortunately, this constraint on u_1 and u_2 is rather inconvenient: It is both *nonlinear* and *implicit*. However, in particular with respect to the fact that we are designing data terms for small displacements, there is an elegant way to overcome this problem: Under the

assumption that the displacements are small and the flow field changes smoothly over time, the left term can be approximated sufficiently well by a first order Taylor expansion in the point $\mathbf{x} = (x_1, x_2, x_3)^\top$:

$$f(x_1 + u_1,\ x_2 + u_2,\ x_3 + 1)\ \approx\ f(x_1, x_2, x_3) + f_{x_1} u_1 + f_{x_2} u_2 + f_t. \qquad (2.3)$$

Then, equation (2.2) simplifies to the following expression:

$$f_{x_1} u_1 + f_{x_2} u_2 + f_t = 0 \qquad (2.4)$$

which is well-known as the *linearised optic flow constraint*. Although this step resolves the problem of non-linearity and implicitness, the obtained constraint still cannot be used directly as data term. Since we are interested in penalising all deviations from zero, we follow the idea of Horn and Schunck and square this constraint thus obtaining the data term [HS81, LK81]

$$M_1(D^1 f, \mathbf{u}) := (\mathbf{u}^\top \nabla_3 f)^2. \qquad (2.5)$$

2.2.1.2   Constancy Assumptions on Image Derivatives

As long as the image data does not violate the brightness constancy assumption, the use of the data term M_1 will give good results. In particular with regard to image data with non-constant brightness, however, constancy assumptions should be based on image features that are less sensitive to illumination changes. A simple and efficient strategy in this context is the consideration of derivatives.

A2 - Constancy of the Gradient. Instead of imposing constancy to the image brightness f from one frame to the next, one may e.g. assume that the spatial brightness gradient $(f_{x_1}, f_{x_2})^\top$ does not change over time [TP84, UGVT88, Sch93, Tis94, ON95, PBB$^+$06, For05]. Since the spatial gradient is a vector with two components, we obtain two constraints this time. They are given by

$$0\ =\ f_{x_1}(x_1 + u_1,\ x_2 + u_2,\ x_3 + 1) - f_{x_1}(x_1, x_2, x_3), \qquad (2.6)$$
$$0\ =\ f_{x_2}(x_1 + u_1,\ x_2 + u_2,\ x_3 + 1) - f_{x_2}(x_1, x_2, x_3). \qquad (2.7)$$

which linearised counterparts

$$\mathbf{u}^\top \nabla_3 f_{x_1}\ =\ 0, \qquad (2.8)$$
$$\mathbf{u}^\top \nabla_3 f_{x_2}\ =\ 0. \qquad (2.9)$$

Squaring and adding them produces the data term

$$M_2(D^2 f, \mathbf{u}) := \sum_{i=1}^{2} (\mathbf{u}^\top \nabla_3 f_{x_i})^2. \qquad (2.10)$$

A3 - Constancy of the Hessian. In a straightforward way, constancy assumptions can also be imposed on higher-order derivatives, e.g. on the (spatial) Hessian $\mathcal{H}_2 f$. Squaring and adding the corresponding equations we obtain the following data term [PBB$^+$06]:

$$M_3(D^3 f, \mathbf{u}) := \sum_{i=1}^{2} \sum_{j=1}^{2} (\mathbf{u}^\top \nabla_3 f_{x_i x_j})^2. \tag{2.11}$$

2.2.1.3 Motion Invariant Constancy Assumptions on Image Derivatives

With M_2 and M_3 we have proposed data terms that are designed for sequences with additive illumination changes. However, they have a small drawback that does not become apparent at first glance: Theoretically, their performance depends on the occurring type of motion. This has the following reason: In contrast to the image brightness both gradient and Hessian contain directional information – in fact, they represent contextual information from a very small neighbourhood. As a consequence, any constancy assumption that is imposed on these expressions implies a constancy assumption on their orientation. On one hand, this property may be useful if it comes to the estimation of translational, divergent or slow rotational motion. In this case the orientation of the features does hardly change and the combination of two or three constraints in one data term may improve the results. On the other hand, slightly poorer results may be obtained if fast rotations are dominating and the implied orientation constancy does not hold.

A4 - Constancy of the Gradient Norm. A way to overcome this theoretical limitation is to create motion invariant image features from these "oriented" derivatives. Instead of imposing constancy on the (spatial) brightness gradient and therewith on its orientation, one may e.g. assume that only its magnitude is constant over time. Then, the following data term is obtained [RD96, PBB$^+$06]:

$$M_4(D^2 f, \mathbf{u}) := (\mathbf{u}^\top \nabla_3 |\nabla f|)^2. \tag{2.12}$$

A5/A6 - Constancy of the Trace/Determinant of the Hessian. This idea to use invariant features can also be extended to higher-order derivatives. As an example, let us consider the (spatial) Hessian $\mathcal{H}_2 f$. In this case, one may either think of imposing constancy on the trace of the Hessian – the (spatial) Laplacian $\Delta_2 f$ – or on the determinant of the (spatial) Hessian $\mathcal{H}_2 f$. While the data term associated to the Laplacian is given by [RD96, PBB$^+$06, WPFS05]:

$$M_5(D^3 f, \mathbf{u}) := (\mathbf{u}^\top \nabla_3 (\Delta_2 f))^2, \tag{2.13}$$

the data term based on the constancy of the determinant of the Hessian reads [PBB$^+$06]

$$M_6(D^3 f, \mathbf{u}) := (\mathbf{u}^\top \nabla_3 \det(\mathcal{H}_2 f))^2. \tag{2.14}$$

This example shows that in general multiple of such scalar valued expressions can be derived from the set of derivatives of a single order. However, there is no general rule which expression gives the best performance.

One should note that all data terms based on derivatives can not only be used in the context of additive illumination changes. If the original image sequence f is replaced by a logarithmised variant $f_{\text{new}} = \log f$, the presented concepts also work fine for sequences with multiplicative variations of the image brightness.

2.2.1.4   Constancy Assumptions on Multiple Image Features

Until now, we have assumed that specific prior knowledge on the scene is available, i.e. that the illumination conditions and the occurring type of motion is known. However, in particular with respect to real-world sequences, this assumption is often not justified. In most cases the required information can only be obtained from sample data and thus lacks completeness. As a consequence, the selection of an appropriate feature for the data term becomes a non-trivial task. In order to tackle this problem, one can make use of the following strategy: Instead of focusing on a single constancy assumption one may combine several possible candidates for the data term by means of a linear combination [Sch93, PBB$^+$06, WPFS05]. Thus, at the expense of additional parameters, the model becomes more flexible - it is not restricted a-priori to sequences of a single type.

A7 - Constancy of the Brightness and the Gradient. In order to give an example for a data term that considers more than one image feature, let us assume that we would like to combine the gradient and the grey value constancy assumption. Then, the following data term is obtained [BBPW04]:

$$M_7(D^1 f, D^2 f, \mathbf{u}) := \gamma_1 \, (\mathbf{u}^\top \nabla_3 f)^2 + \gamma_2 \sum_{i=1}^{2} (\mathbf{u}^\top \nabla_3 f_{x_i})^2. \qquad (2.15)$$

Here, γ_1 and γ_2 are positive weights that steer the influence of the corresponding constancy assumptions on the overall data term.

One should note that the proposed strategy may also be applied in the context of image sequences *with* prior knowledge: In this case the combination of several (appropriate) image features may give additional useful information compared to a data term based on a single feature only. Consequently, the accuracy of the computed displacement field may improve. As we will see in Section 2.2.3.3, this aspect is also related to the so-called aperture problem [BPT88].

2.2.1.5   Constancy Assumptions on Multiple Image Channels

By the example of M_7 we have shown how to design data terms that can handle constancy assumptions on multiple image features. Regarding the input data, however, we still have restricted ourselves to image sequences with a single information channel (e.g. grey value image sequences). Since most of todays cameras provide colour images, let us now lift this limitation and discuss suitable strategies for generalising single-channel data terms to a scenario with multi-channel image data.

Tab. 2.1: Comparison of the data terms M_1–M_8.

	data term	constancy assumption	illum. changes	motion type
M_1	$(\mathbf{u}^\top \nabla_3 f)^2$	brightness	no	any
M_2	$\sum_{i=1}^{2}(\mathbf{u}^\top \nabla_3 f_{x_i})^2$	gradient	yes	translational divergent slow rotational
M_3	$\sum_{i=1}^{2}\sum_{j=1}^{2}(\mathbf{u}^\top \nabla_3 f_{x_i x_j})^2$	Hessian	yes	translational divergent slow rotational
M_4	$(\mathbf{u}^\top \nabla_3 \lvert \nabla f \rvert)^2$	gradient magnitude	yes	any
M_5	$(\mathbf{u}^\top \nabla_3 (\Delta_2 f))^2$	Laplacian	yes	any
M_6	$(\mathbf{u}^\top \nabla_3 \det(\mathcal{H}_2 f))^2$	Hessian determinant	yes	any
M_7	$\gamma_1 (\mathbf{u}^\top \nabla_3 f)^2$ $+$ $\gamma_2 \sum_{i=1}^{2}(\mathbf{u}^\top \nabla_3 f_{x_i})^2$	brightness $+$ gradient	depends on γ_1 and γ_2	
M_8	$\sum_{i=1}^{3}(\mathbf{u}^\top \nabla_3 f_{(i)})^2$	RGB colour brightness	no	any

When considering image sequences with multiple channels, one is evidently interested in exploiting all available information. Therefore, a simple conversion of the data by combining all channels to a scalar-valued image sequence is definitely not the best solution. Instead it makes much more sense to treat all channels separately and recombine the resulting constraints in a common data term. This does not only allow to preserve the additional information, it also enforces the computation of a joint displacement field for all channels. Let us illustrate this proceeding by the example of the brightness constancy assumption.

A8 - Constancy of the Brightness for Multiple Channels. Let $\mathbf{f}(x_1, x_2, x_3) : \Omega_2 \times T \to \mathbb{R}^n$ be a multi-channel image sequence with n different channels $f_{(1)}, ..., f_{(n)}$. Then, the data term for the generalised brightness constancy assumption reads

$$M_8(D^1 f_{(1)}, ..., D^1 f_{(n)}, \mathbf{u}) := \sum_{i=1}^{n} \mu_i (\mathbf{u}^\top \nabla_3 f_{(i)})^2 \qquad (2.16)$$

Here, μ_i are positive weights that steer the influence of the different channels.

One should note that in the case of RGB sequences these weights are often set to 1. This intuitively makes sense, since the red, blue and green channel should have similar impact on the result. If the channels, however, are not directly comparable in their meaning (e.g. YUV images) or have been obtained from different sources (e.g. from colour and infrared sensors) then adjusting these weights adequately may significantly improve the results.

An overview of all data terms presented so far is given in Table 2.1. Thereby one should note that data terms based on higher orders of image derivatives are in general more sensitive to noise than assumptions based on the original data itself.

2.2.2   Constancy Assumptions - Experiments

After we have discussed a variety of possible constancy assumptions, let us now illustrate their impact on the computed flow field by means of two experiments. To this end, we consider the data terms M_1–M_8 within a spatial energy functional based on homogeneous regularisation of Horn–Schunck type, i.e. we minimise

$$E_j(\mathbf{u}) = \int_{\Omega_2} \left(M_j + \alpha \sum_{i=1}^{2} |\nabla u_i|^2 \right) d\mathbf{x} \tag{2.17}$$

for $j = 1,...,8$. In order to allow for a quantitative comparison of the results we computed the so-called *average angular error (AAE)* [BFB94] in all cases. It can be calculated via

$$\text{AAE}(\mathbf{u}_c, \mathbf{u}_e) = \frac{1}{|\Omega_2|} \int_{\Omega_2} \arccos \left(\frac{\mathbf{u}_c^\top \mathbf{u}_e}{|\mathbf{u}_c||\mathbf{u}_e|} \right) d\mathbf{x}, \tag{2.18}$$

where the subscripts c and e denote the correct respectively the estimated spatiotemporal optic flow vectors $\mathbf{u}_c = (u_{c1}, u_{c2}, 1)^\top$ and $\mathbf{u}_e = (u_{e1}, u_{e2}, 1)^\top$. In this context, $|\Omega_2| = \int_{\Omega_2} d\mathbf{x}$ stands for the integration domain, and $|.|$ denotes the Euclidean norm given by $|\mathbf{u}| = \sqrt{u_1^2 + u_2^2 + 1}$. For a qualitative evaluation of the computed flow fields we have used the colour representation shown in Figure 2.2. While the colour itself indicates the direction of the displacements, the brightness expresses their magnitude.
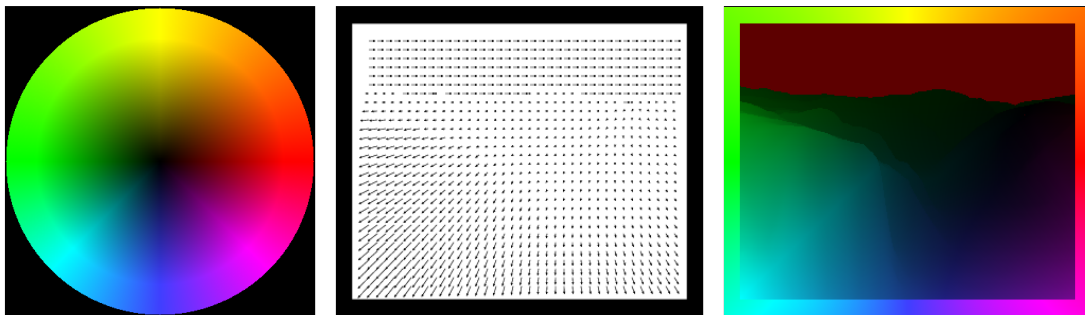
Fig. 2.2:   **(a) Left:** Colour code. **(b) Centre:** Flow field in vector representation. **(c) Right:** Corresponding colour representation (boundaries contain the colour code).

2.2.2.1 Experiment I: Translational and Divergent Motion, Varying Illumination

In our first experiment depicted in Figure 2.3 we consider the famous *Yosemite* sequence *with* clouds. It consists of 15 frames of size 316×252 pixels and combines divergent and translational motion under varying illumination. Both the sequence and its ground truth flow field are available from `ftp://csd.uwo.ca` under the directory `pub/vision`. For evaluating the performance of the colour constancy assumption (M_8), we also created a RGB variant with orange mountain-site and blue sky. While the chosen colours reflect additional knowledge on the segmentation of the final flow field, they have the same grey values as the original sequence (if converted back).

The obtained results for optimised Gaussian presmoothing parameter σ (cf. equation (2.1)) and regularisation parameter α are presented in Table 2.2. Moreover, in the case of the combined data term M_7, also the optimised values for the weights γ_1 and γ_2 are listed. Comparing the achieved angular errors, the following three observations can be made:

- *Usefulness of Image Derivatives under Varying Illumination:* Firstly, we can see that the commonly used brightness constancy assumption is outperformed by almost all other constraints that involve higher derivatives. This is not surprising, since the sequence was chosen in such a way, that illumination changes are present. The poor performance of the brightness constancy assumption is also reflected in the result for the combined data term M_7. Here, the average angular error is only marginally better than for the pure gradient constancy assumption M_2.

- *Usefulness of Directional Information for Translational and Divergent Motion:* Secondly, if we compare the results for the gradient constancy assumption and the gradient norm constancy assumption, one can see that the additional directional information helps to improve the accuracy of the estimation In this context we also observe that constancy assumptions based on higher order derivatives require a larger standard value for the standard deviation σ in order to give optimal results. This reflects the increased sensitivity of the underlying constraints to the influence of noise. In contrast, the significantly varying values for α have a much simpler reason. They are simple caused by the different range of values of the different image features.

- *Usefulness of Colour Information:* Finally, when comparing the average angular error for the colour and the grey value constancy assumption, one can observe a slight improvement of the result. This is a consequence of the additional colour information, that allows a slightly better distinction of the two motion patterns in the mountain region and the sky than in the case of the original grey value sequence. Thereby one should keep in mind that, due to the occurring illumination changes in the region of the clouds, assumptions on the original grey/colour values of the image sequence are not optimal.

These quantitative impressions are confirmed qualitatively by the corresponding flow fields shown in Figure 2.3. While M_1 and M_8 give slightly better results at the mountain site, the other data terms are significantly superior in estimating the sky region where illumination changes are present. This shows once more that it can be worthwhile to replace the brightness constancy constraint by constraints that involve higher derivatives, in particular when varying illumination has to be expected.
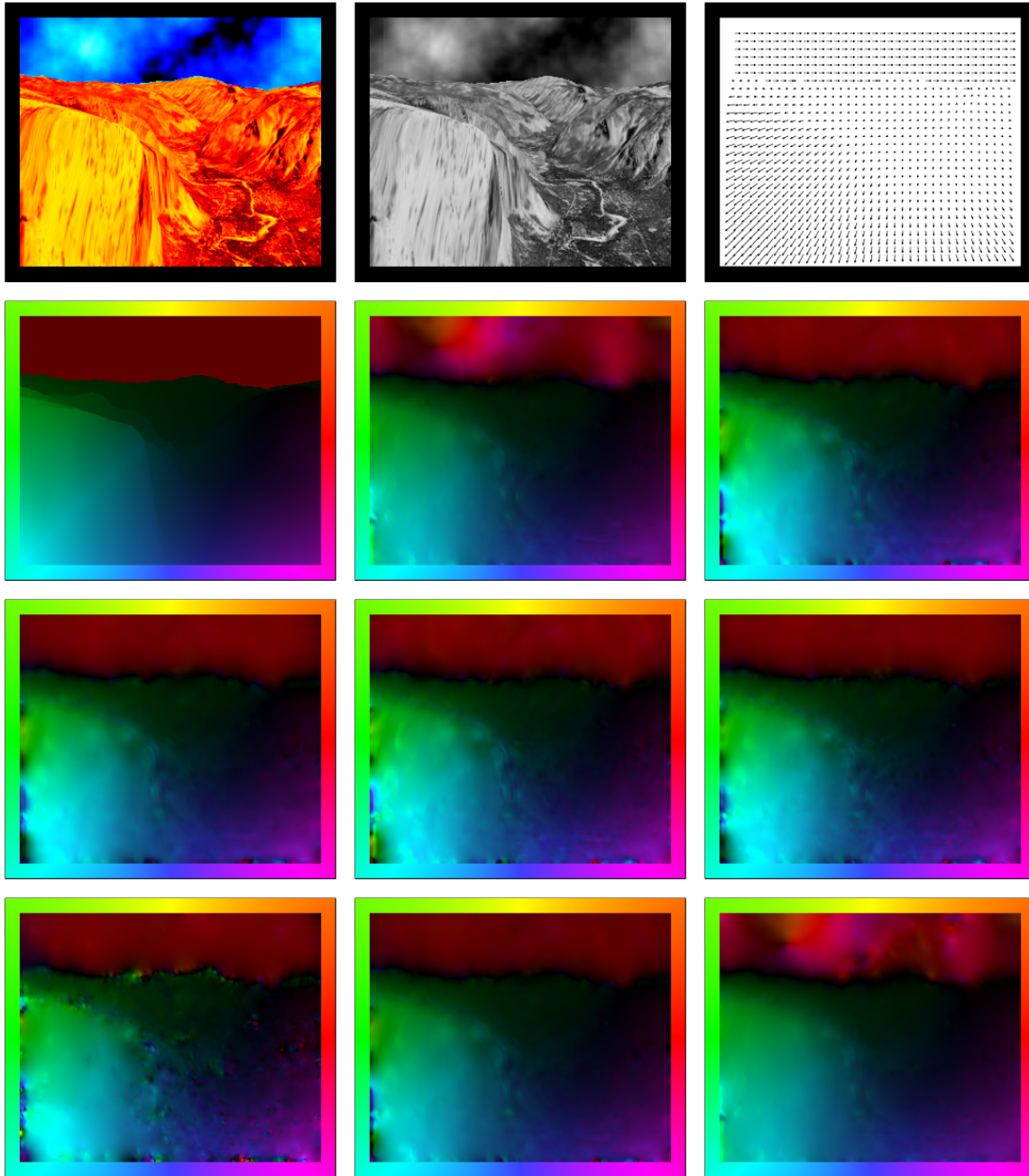
Fig. 2.3: **Experiment I**: Impact of the constancy assumption on the quality of the flow field for *translational* and *divergent motion* with *varying illumination*. **(a) Top Left:** Frame 8 of the *Yosemite* sequence *with* clouds of size 316×256 pixels (colour variant). **(b) Top Centre:** The same frame of the original sequence (grey-scale variant). **(c) Top Right:** Ground truth (vector plot). **(d) Upper Middle Left:** Ground truth (colour plot). **(e) Upper Middle Centre:** Computed flow field for a spatial approach with data term M_1 (brightness constancy) and homogeneous regularisation. **(f) Upper Middle Right:** Data term M_2 (gradient constancy). **(g) Lower Middle Left:** Data term M_3 (constancy of Hessian). **(h) Lower Middle Centre:** Data term M_4 (gradient magnitude constancy). **(j) Lower Middle Right:** Data term M_5 (constancy of Laplacian). **(j) Bottom Left:** Data term M_6 (constancy of Hessian determinant). **(k) Bottom Centre:** Data term M_7 (brightness and gradient constancy). **(l) Bottom Right:** Data term M_8 (colour constancy).

Tab. 2.2: **Experiment I**: Impact of the constancy assumption on the quality of the optic flow field for *translational* and *divergent motion* with *varying illumination*. We used a spatial energy functional with homogeneous regularisation, and computed the average angular error (AAE) for the *Yosemite* sequence *with* clouds. The parameters σ, α, γ_1 and γ_2 have been optimised.

data term	constancy assumption	σ	α	AAE
M_1	brightness	1.30	500	7.17°
M_2	gradient	2.10	20	5.91°
M_3	Hessian	2.70	1.8	6.46°
M_4	gradient magnitude	1.90	14	6.37°
M_5	Laplacian	2.50	3.0	6.18°
M_6	Hessian determinant	3.00	0.1	8.10°
M_7	brightness + gradient ($\gamma_1 = 0.01$,$\gamma_2 = 1$)	2.10	21	5.87°
M_8	RGB colour brightness	1.20	2400	7.10°

2.2.2.2 Experiment II: Rotational Motion

For our second experiment we chose the *Slowly Rotating Texture* sequence shown in Figure 2.4. It consists of 10 frames of size 127×127 pixels and depicts a green textured disc that rotates in front of a blue background with similar texture. The angle of rotation between two consecutive frames of the sequence is given by 5 degrees. Therefore, as desired, the displacements remain relatively small – at least at locations that are sufficiently close to the centre.

As for our first experiment, the optimised results with respect to the parameters σ, α, γ_1 and γ_2 are listed in a table (Table 2.2). Here, the following two observations can be made:

- *Not Necessarily Advantages by Motion-Invariant Image Features:* Firstly, we can see that the data terms based on motion-invariant image features are not necessarily better than those based on directional information. Although this observation may be surprising at first glance, it has a simple explanation: Since the angle of rotation is only five degrees, the property of motion-invariance does not play a significant role. In fact, in the case of this heavily textured sequence, it is more important to consider structural (neighbourhood) information. This can simply be done by using assumptions on derivatives. As a consequence, the classical data term based on the image brightness is once again outperformed by all other constraints.

- *Usefulness of Colour Information:* Secondly, one can observe that the additional information provided by the colour version of the image sequence is seemingly very useful. Comparing the images of the original and the grey-value sequence, one sees that the rotating disc is hardly distinguishable from the background if no colour information is present. Since moreover, the motion discontinuity coincides with the object boundary, this information is particularly helpful to improve the results (correspondences within one colour a favoured).

Also in this case, most of our considerations are confirmed by the corresponding flow fields. Evidently, all data terms are capable of estimating the displacements close to the
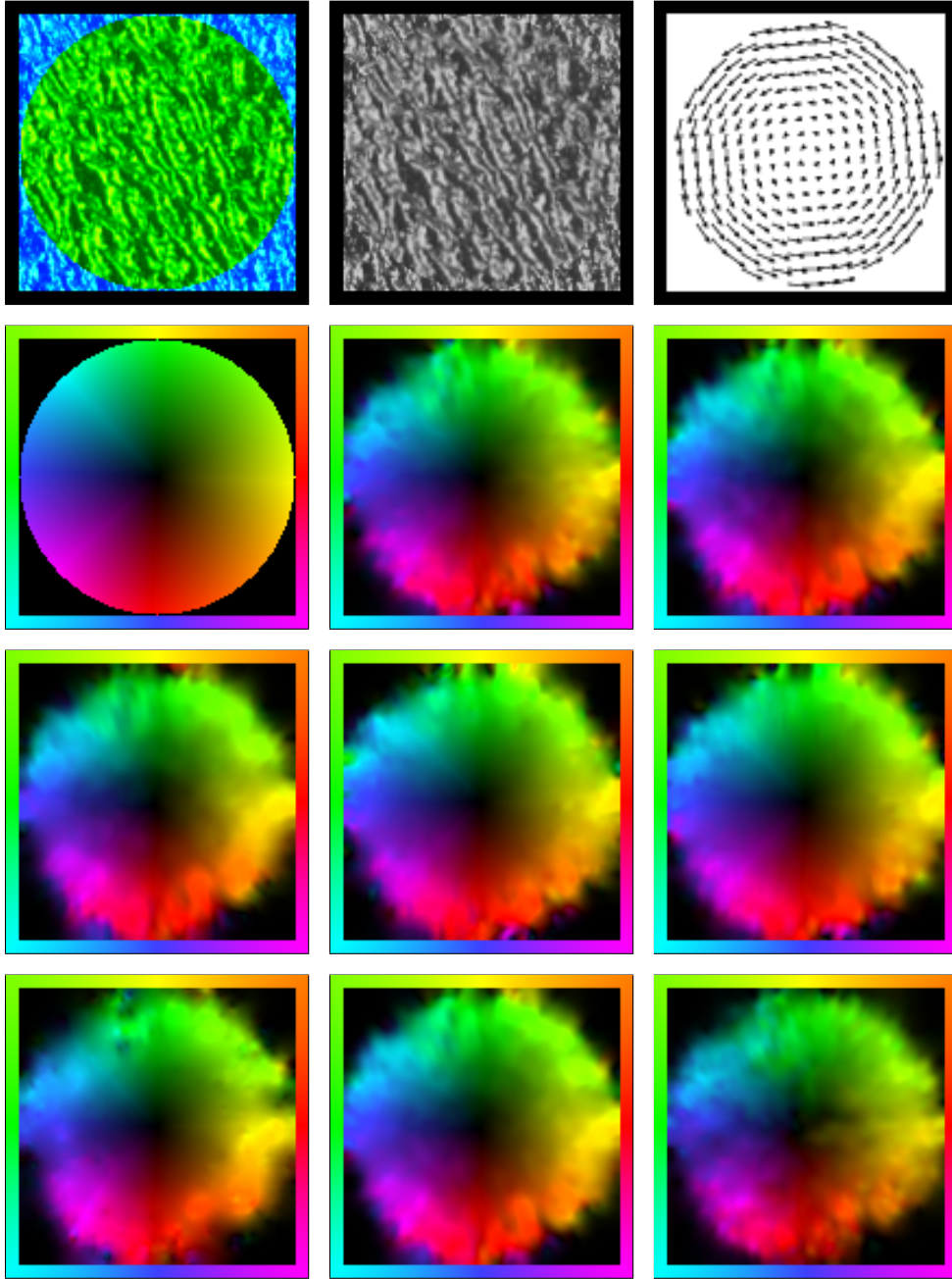
Fig. 2.4:   **Experiment II**: Impact of the constancy assumption on the quality of the flow field for *rotational motion*. **(a) Top Left:** Frame 2 of the *Slowly Rotating Texture* sequence of size 127×127 pixels (colour variant). **(b) Top Centre:** The same frame of the converted sequence (grey-scale variant). **(c) Top Right:** Ground truth (vector plot). **(d) Upper Middle Left:** Ground truth (colour plot). **(e) Upper Middle Centre:** Computed flow field for a spatial approach with data term M_1 (brightness constancy) and homogeneous regularisation. **(f) Upper Middle Right:** Data term M_2 (gradient constancy). **(g) Lower Middle Left:** Data term M_3 (constancy of Hessian). **(h) Lower Middle Centre:** Data term M_4 (gradient magnitude constancy). **(j) Lower Middle Right:** Data term M_5 (constancy of Laplacian). **(j) Bottom Left:** Data term M_6 (constancy of Hessian determinant). **(k) Bottom Centre:** Data term M_7 (brightness and gradient constancy). **(l) Bottom Right:** Data term M_8 (colour constancy).

Tab. 2.3: **Experiment II**: Impact of the constancy assumption on the quality of the optic flow field for *rotational motion*. We used a spatial energy functional with homogeneous regularisation, and computed the average angular error (AAE) for the *Slowly Rotating Texture* sequence in Figure 2.4. The parameters σ, α, γ_1 and γ_2 have been optimised.

data term	constancy assumption	σ	α	AAE
M_1	brightness	2.60	50	12.15°
M_2	gradient	3.50	1.8	11.47°
M_3	Hessian	3.50	0.5	11.93°
M_4	gradient magnitude	3.80	0.5	11.84°
M_5	Laplacian	4.60	0.05	11.68°
M_6	Hessian determinant	3.30	0.5	11.74°
M_7	brightness + gradient ($\gamma_1 = 0.05$,$\gamma_2 = 1$)	3.10	100	11.26°
M_8	RGB colour brightness	2.10	410	10.62°

centre in an accurate way. However, with increasing distance to the center, the quality of the estimation becomes worse since the displacements get larger. In particular, at the motion boundary, the differences between the approaches can be seen. Thereby, the data term based on the colour information gives also visually the best impression (hardly overestimates).

2.2.3 Constancy Assumptions - The Motion Tensor Notation

In Section 2.2.1, we have formulated any data term based on one or multiple linear constraints as a (weighted) sum of inner products between the spatiotemporal flow vector $\mathbf{u} = (u_1, u_2, 1)^\top$ and the spatiotemporal gradient of the corresponding image features that are assumed to be constant over time. In order to simplify the notation further and in order to understand the local properties of the different data terms in a better way, let us now extend this formalism by introducing the concept of *motion tensors* [BGW91, Jäh93, SAH91, Far01, BWKS06].

2.2.3.1 The Motion Tensor for One Constraint

Let us first illustrate the concept of motion tensors by the example of the brightness constancy assumption. Since this assumption is based on a single constraint only, it can easily be reformulated as the following quadratic form:

$$M_1(D^1 f, \mathbf{u}) = (\mathbf{u}^\top \nabla_3 f)^2 = \mathbf{u}^\top \, \nabla_3 f \nabla_3 f^\top \, \mathbf{u} =: \mathbf{u}^\top \, J_1(\nabla_3 f) \, \mathbf{u}. \qquad (2.19)$$

Evidently, the *symmetric positive semidefinite* 3×3 matrix $J_1(\nabla_3 f) := \nabla_3 f \nabla_3 f^\top$ contains all necessary information to describe the constraint on the local solution imposed by M_1. Therefore, in the following, we will refer to it as the *motion tensor* that is associated to the data term M_1. In fact, such a motion tensor can be derived in the same way also for all other data terms based on a single constraint (e.g. M_4, M_5 and M_6).

2.2.3.2   The Motion Tensor for Multiple Constraints

After we have seen how to construct motion tensors for data terms with a single constraint, let us now generalise this idea to data terms with multiple constraints. This extended case applies to the data terms M_2, M_3, M_7 and M_8. To this end, we consider an arbitrary data term that can be represented by the weighted sum of n squared linearised constraints on the image features $p_1, ..., p_n$. Then, the corresponding motion tensor can be obtained from the quadratic form

$$
\sum_{i=1}^{n} \gamma_i \left(\mathbf{u}^\top \nabla_3 p_i \right)^2 \;=\; \sum_{i=1}^{n} \gamma_i \left(\mathbf{u}^\top \nabla_3 p_i \nabla_3 p_i^\top \mathbf{u} \right) = \sum_{i=1}^{n} \gamma_i \left(\mathbf{u}^\top \underbrace{J(\nabla_3 p_i)}_{\text{constraint tensor}} \mathbf{u} \right)
$$

$$
=\; \mathbf{u}^\top \left(\sum_{i=1}^{n} \gamma_i \, J(\nabla_3 p_i) \right) \mathbf{u} = \mathbf{u}^\top \underbrace{J(\nabla_3 p_1, .., \nabla_3 p_n)}_{\text{general motion tensor}} \mathbf{u},
$$

$$(2.20)$$

where $\gamma_1, ..., \gamma_n$ are the (positive) weights of the different constraints.

As one can see, each of the constraints contributes with exactly one matrix (constraint tensor) to the common motion tensor. Since each of these 3×3 matrices is *symmetric positive semidefinite* by construction and this property is preserved under weighted averaging, it is also shared by the general motion tensor.

One should note that from a statistical point of view such combined data terms are nothing else than *ordinary least squares fits* (OLS) with respect to the considered constancy assumptions. This can be easily verified by reformulating them as

$$
\sum_{i=1}^{n} \gamma_i \left(\mathbf{u}^\top \nabla_3 p_i \right)^2 \;=\; \sum_{i=1}^{n} \gamma_i \left((u_1, u_2, 1) \, (p_{i_{x_1}}, p_{i_{x_2}}, p_{i_{x_3}})^\top \right)^2 \tag{2.21}
$$

$$
=\; \sum_{i=1}^{n} \left((u_1, u_2) \, (\sqrt{\gamma_i}\, p_{i_{x_1}}, \sqrt{\gamma_i}\, p_{i_{x_2}})^\top - (- \sqrt{\gamma_i}\, p_{i_{x_3}}) \right)^2
$$

$$
=\; \underbrace{\left| \begin{pmatrix} \sqrt{\gamma_1}\, p_{1_{x_1}} & \sqrt{\gamma_1}\, p_{1_{x_2}} \\ \vdots & \vdots \\ \sqrt{\gamma_n}\, p_{n_{x_1}} & \sqrt{\gamma_n}\, p_{n_{x_2}} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} - \begin{pmatrix} - \sqrt{\gamma_1}\, p_{1_{x_3}} \\ \vdots \\ - \sqrt{\gamma_n}\, p_{n_{x_3}} \end{pmatrix} \right|^2}_{|A\tilde{\mathbf{u}} - \mathbf{b}|^2}.
$$

2.2.3.3   The Local Interaction Between Data and Smoothness Term

So far the motion tensor notation did only allow for a compact formulation of the data term. However, it may also serve as a tool to obtain information on the local interaction between the data and the smoothness term. In order to understand this interaction, it is important to know if the local information of the data term is sufficient to determine the two unknowns u_1 and u_2, at a point (x_1, x_2, x_3) uniquely or not. Let us therefore first discuss how the local minimiser of the data term can be computed and then how the motion tensor notation can provide us with a reliable indicator for its uniqueness.

The Local Minimiser of the Data Term. In order to determine the local solution of the data term, one has to minimise it. This can be done by setting its partial derivatives with respect to u_1 and u_2 to zero and by solving the resulting 2×2 system of equations. For a general data term of type (2.20) this system is given by

$$\underbrace{\begin{pmatrix} \sum_{i=1}^{n} \gamma_i\, p_{i_{x_1}}^2 & \sum_{i=1}^{n} \gamma_i\, p_{i_{x_1}} p_{i_{x_2}} \\ \sum_{i=1}^{n} \gamma_i\, p_{i_{x_1}} p_{i_{x_2}} & \sum_{i=1}^{n} \gamma_i\, p_{i_{x_2}}^2 \end{pmatrix}}_{A^\top A} \underbrace{\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}}_{\tilde{\mathbf{u}}} = \underbrace{\begin{pmatrix} -\sum_{i=1}^{n} \gamma_i\, p_{i_{x_1}} p_{i_{x_3}} \\ -\sum_{i=1}^{n} \gamma_i\, p_{i_{x_2}} p_{i_{x_3}} \end{pmatrix}}_{A^\top \mathbf{b}}, \quad (2.22)$$

where the resulting equations are referred to as the so-called *normal equations* [SB02]. If we denote by $J_{nm}(\nabla_3\, p_1, .., \nabla_3\, p_n)$ the entry of the general motion tensor at position (n, m) this system can be reformulated as

$$\begin{pmatrix} J_{11}(\nabla_3\, p_1, .., \nabla_3\, p_n) & J_{12}(\nabla_3\, p_1, .., \nabla_3\, p_n) \\ J_{12}(\nabla_3\, p_1, .., \nabla_3\, p_n) & J_{22}(\nabla_3\, p_1, .., \nabla_3\, p_n) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} -J_{13}(\nabla_3\, p_1, .., \nabla_3\, p_n) \\ -J_{23}(\nabla_3\, p_1, .., \nabla_3\, p_n) \end{pmatrix}. \quad (2.23)$$

Since this 2×2 matrix carries all *spatial* information of the the full motion tensor, we will denote it in the following as the *spatial motion tensor*. Evidently, this system can only be solved uniquely, if this spatial motion tensor is invertible.

Interpretation of the Eigenvalues. In order to determine if the spatial motion tensor is invertible and in order to obtain some additional information on the local solution of the underlying data term, let us take a look at the eigenvalue decomposition of $A^\top A$. Let such a decomposition be given by

$$\begin{pmatrix} J_{11}(\nabla_3\, p_1, .., \nabla_3\, p_n) & J_{12}(\nabla_3\, p_1, .., \nabla_3\, p_n) \\ J_{12}(\nabla_3\, p_1, .., \nabla_3\, p_n) & J_{22}(\nabla_3\, p_1, .., \nabla_3\, p_n) \end{pmatrix} = (\mathbf{v_1}, \mathbf{v_2}) \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} (\mathbf{v_1}, \mathbf{v_2})^\top,$$
$$(2.24)$$

where λ_1 and λ_2 denote the eigenvalues (w.l.o.g. let $\lambda_1 \geq \lambda_2$) and $\mathbf{v_1}$ and $\mathbf{v_2}$ are the corresponding orthonormal eigenvectors. Then, three different cases of solutions of the data term can be distinguished which depend on the *rank* of this tensor:

- *Rank 2 – Optimal Rank.* The spatial motion tensor is invertible and the data term offers a locally unique solution. Thereby the two non-zero eigenvalues indicate that at least two of the image features $p_1, ..., p_n$ vary in different spatial directions. This in turn means that the corresponding spatial gradients $\nabla_2 p_i$ and $\nabla_2 p_j$ of these two features are *linearly independent*. Since each of these constraint allows to recover exactly the flow information parallel to its spatial feature gradient (aperture problem), these two linearly independent constraints are sufficient to determine the two unknowns u_1 and u_2 uniquely. The corresponding solution is then given by

$$\tilde{\mathbf{u}} = (A^\top A)^{-1} A^\top \mathbf{b}. \quad (2.25)$$

This case is illustrated in Figure 2.5 (a). Here two features – the horizontal and the vertical line – provide a unique solution of the data term.

- *Rank 1 – Rank Deficiency of 1.* The spatial motion tensor is not invertible. Since at least one solution of the underlying least square fit must exist, this means that the data term has infinitely many solutions and is thus locally non-unique. Thereby the fact that only one eigenvalue is non-zero shows that all spatial feature gradients $\nabla_2 p_1, .., \nabla_2 p_n$ are *locally dependent*, but at least one of them gives information i.e. $\exists i \in \{1, ., n\} : \nabla_2 p_i \neq 0$. Exactly in direction of this non-zero feature gradient, the component of the optic flow can be computed. It is given by

$$\tilde{\mathbf{u}} = (A^\top A)^+ A^\top \mathbf{b}, \tag{2.26}$$

  where $(A^\top A)^+$ is the Moore-Penrose pseudoinverse [HJ94] of $(A^\top A)$. While the component in direction of the non-zero feature gradient can be computed, the component orthogonal to it cannot be determined. This results in an *ambiguity* of the solution of rank-1 spatial motion tensors *along a line*. This case is illustrated in Figure 2.5 (b). Since only one image feature – the vertical line – is available, not more than the flow component in direction of the corresponding feature gradient can be computed (represented by the green arrow). However, also all other displacements that map the left point to one of the points on the right line are valid local solutions of the data term.

  One should note that the classical aperture problem resulting from the brightness constancy assumption that was discussed in the introduction is only a special case of this scenario: If only one constraint is used, it is evident that this constraint cannot be sufficient to determine the two unknowns u_1 and u_2 uniquely. The component of the optic flow that can be computed in this case – the component in direction of the non-zero brightness gradient – is known as the *normal flow*.

- *Rank 0 – Rank Deficiency of 2.* The spatial motion tensor is not invertible. As in the previous case the data term has infinitely many solutions and is thus locally non-unique. Since both eigenvalues are zero, this means that there are no spatial variations of the image features, i.e. $p_{i_{x_1}} = 0$ and $p_{i_{x_2}} = 0$. As a consequence, all entries of the spatial motion tensor and the right hand side become zero. This in turn means that all values for u_1 and u_2 are local solutions of the data term. Thus, we have an *ambiguity* of the solution of rank-0 spatial motion tensors *in the whole image plane*. This case is illustrated in Figure 2.5 (c). Since no information is available, the correspondences may lie everywhere.

The preceding analysis of the eigenvalues shows the close relation between the rank of the spatial motion tensor and the uniqueness of the solution. Moreover, it allows to interpret the corresponding space of solutions geometrically. However, this analysis has also one drawback: In the case of a unique solution it does not give sufficient information to decide on its reliability. It does not distinguish solutions that are in accordance with all local constraints and thus yield a very small contribution of the data term to the whole energy functional and solutions that violate one or more constancy assumptions severely and thus yield to large local contribution. In order to obtain this information, one has to compute the local solution first and then to use it to evaluate the local contribution of the data term.
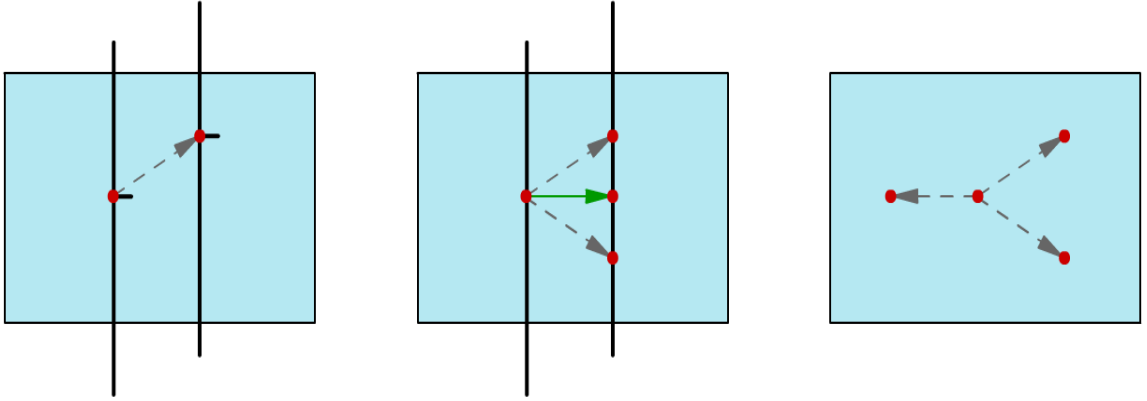
Fig. 2.5: Visualisation of the three cases for the solution of the data term depending of the rank of the spatial motion tensor. **(a) Left:** Optimal Rank. Unique solution determined by two linearly independent feature gradients (horizontal and vertical line). **(b) Centre:** Rank Deficiency of 1. Non-unique solution due to one feature gradient only (vertical line). At least the normal component is computable (green arrow). **(c) Right:** Rank Deficiency of 2. No information is available. Correspondences may lie everywhere.

This shows that for a better understanding of the contribution of the data term the rank of the spatial motion tensor is very important. Unfortunately, the actual rank of this matrix depends on the local structure of the image sequence and thus cannot be computed in advance. However, it is at least possible to determine its maximum rank. In most cases this gives already a good impression on the behaviour of the data term.

Relevance of the Rank Deficiency Problem for Variational Approaches. Of course, variational approaches do not suffer from the rank deficiency problem: They regularise a possibly non-unique local solution of the data term by additionally imposing (piecewise) smoothness of the result. Nevertheless, the information given by the rank of the spatial motion tensor is still very useful: It determines, whether the regulariser is locally dominating and if the data term can be fulfilled in accordance with the smoothness constraint. While data terms based on spatial motion tensors with a maximum rank of 1 can sometimes be fulfilled without contradicting the smoothness term (e.g. if both constraints are orthogonal [NE86]), data terms based on spatial motion tensors with rank 2 may provide their own solution and thus seek for a real compromise with the regulariser.

An overview of the different motion tensors associated to the data terms presented so far as well as the maximum rank of their spatial motion tensors are given in Table 2.4.

2.2.4 Constancy Assumptions - Experiments II

In the previous section we have discussed the importance of the rank of the spatial motion tensor for the local interaction between the data and the smoothness term. Unfortunately, we could only compare the proposed motion tensors by the *maximum* rank of their spatial counterparts. For a specific image sequence, however, we can go one step further and evaluate their *actual* rank. This shall be done in the next experiment.

Tab. 2.4: Comparison of the motion tensors J_1–J_8 and the rank of their spatial motion tensors.

	motion tensor	constancy assumption	rank
J_1	$\nabla_3 f \nabla_3 f^\top$	brightness	≤ 1
J_2	$\sum\limits_{i=1}^{2} (\nabla_3 f_{x_i})(\nabla_3 f_{x_i})^\top$	gradient	≤ 2
J_3	$\sum\limits_{i=1}^{2}\sum\limits_{j=1}^{2} (\nabla_3 f_{x_i x_j})(\nabla_3 f_{x_i x_j})^\top$	Hessian	≤ 2
J_4	$\dfrac{(f_{x_1}\nabla_3 f_{x_1}+f_{x_2}\nabla_3 f_{x_2})(f_{x_1}\nabla_3 f_{x_1}+f_{x_2}\nabla_3 f_{x_2})^\top}{f_{x_1}^2+f_{x_2}^2}$	gradient magnitude	≤ 1
J_5	$(\nabla_3 \sum\limits_{i=1}^{2} f_{x_i x_i})(\nabla_3 \sum\limits_{i=1}^{2} f_{x_i x_i})^\top$	Laplacian	≤ 1
J_6	$(f_{x_2 x_2}\nabla_3 f_{x_1 x_1}+f_{x_1 x_1}\nabla_3 f_{x_2 x_2}-2f_{x_1 x_2}\nabla_3 f_{x_1 x_2})$ $(f_{x_2 x_2}\nabla_3 f_{x_1 x_1}+f_{x_1 x_1}\nabla_3 f_{x_2 x_2}-2f_{x_1 x_2}\nabla_3 f_{x_1 x_2})^\top$	Hessian determinant	≤ 1
J_7	$\gamma_1\ \nabla_3 f \nabla_3 f^\top + \gamma_2 \sum\limits_{i=1}^{2} (\nabla_3 f_{x_i})(\nabla_3 f_{x_i})^\top$	brightness + gradient	≤ 2
J_8	$\sum\limits_{i=1}^{3} (\nabla_3 f_{(i)})(\nabla_3 f_{(i)})^\top$	RGB colour brightness	≤ 2

2.2.4.1   Experiment I: The Rank of the Spatial Motion Tensor

In this experiment, we compare the *actual* rank of the spatial motion tensors associated to J_1-J_8 for the *Rotating Sphere* sequence depicted in Figure 2.6. This sequence consists of 20 frames of size 150×150 and shows a rotating sphere in front of a stationary background. The texture of the background is thereby chosen in such a way that it only changes in one direction and the aperture problem is present. The sequence as well as its ground truth are

available at the internet address `www.cs.otago.ac.nz/research/vision/`. The computed rank maps for all spatial motion tensors are presented in Figure 2.6. As treshold for both eigenvalues λ_1 and λ_2 a treshold of 0.01 was used. In the case of unique solution (rank 2), we moreover also distinguished between the case, where the data term is close to zero (green) and where the constancy assumptions are locally violated (red). The depicted ranks maps show the following observations:

- *Low Textured Areas Require Regularisation.* As one can see, all constancy assumptions have problems to overcome the rank deficiency problem in the background. In particular, the spatial motion tensor associated to the data term M_6 (constancy of the determinant of the Hessian) is almost everywhere of rank zero. This shows, that the determinant of the Hessian is a very sparse feature. This is intuitively clear, since this feature already becomes zero if one eigenvalue of the Hessian is zero (not to confuse with the eigenvalues of the motion tensor).

- *Usefulness Of Additional Constraints.* Moreover, the presented rank maps show that motion tensors of rank 2 really provide additional information: They can overcome the rank deficiency problem in almost the whole sphere. Only at a very few locations at the boundary of sphere the constancy assumptions are actually violated. This, however, is evident, since due to the rotational motion new parts of the sphere appear and disappear.

This experiment demonstrates that the consideration of multiple constraints can be worthwhile in certain situations. However, it also shows that additional constraints cannot provide additional information, where no no additional information is available (e.g. in the background).

2.2.5   Robust Data Terms - Modelling

After we have chosen one or more suitable constancy assumptions for our task, let us now discuss by the example of M_1 (brightness constancy) and M_7 (brightness and gradient constancy) how the corresponding data terms can be modified such that they become more robust. To this end, we investigate three different strategies: *local least squares fitting*, *adaptive averaging with nonlinear diffusion* and *nonquadratic penalisation*. While the first two concepts improve the performance for corrupted data (e.g. for noisy image sequences), the second one renders the estimation more robust with respect to violations of the model assumptions (e.g. with respect to appearing or occluded objects).

2.2.5.1   Local Least Squares Fitting

A useful strategy to make optic flow estimation more robust under noise is the consideration of neighbourhood information within the data term [BWS02, BWS05]. To this end, one may e.g. assume that the optic flow is constant within some spatial or spatiotemporal neighbourhood of fixed size. Then, simple statistical methods such as least square regressions can be applied to estimate the flow vector from the considered neighbourhood [LK81]. This yields so-called *combined local global* (CLG) methods [BWS02] that combine the robustness of local approaches with the density of global variational approaches.
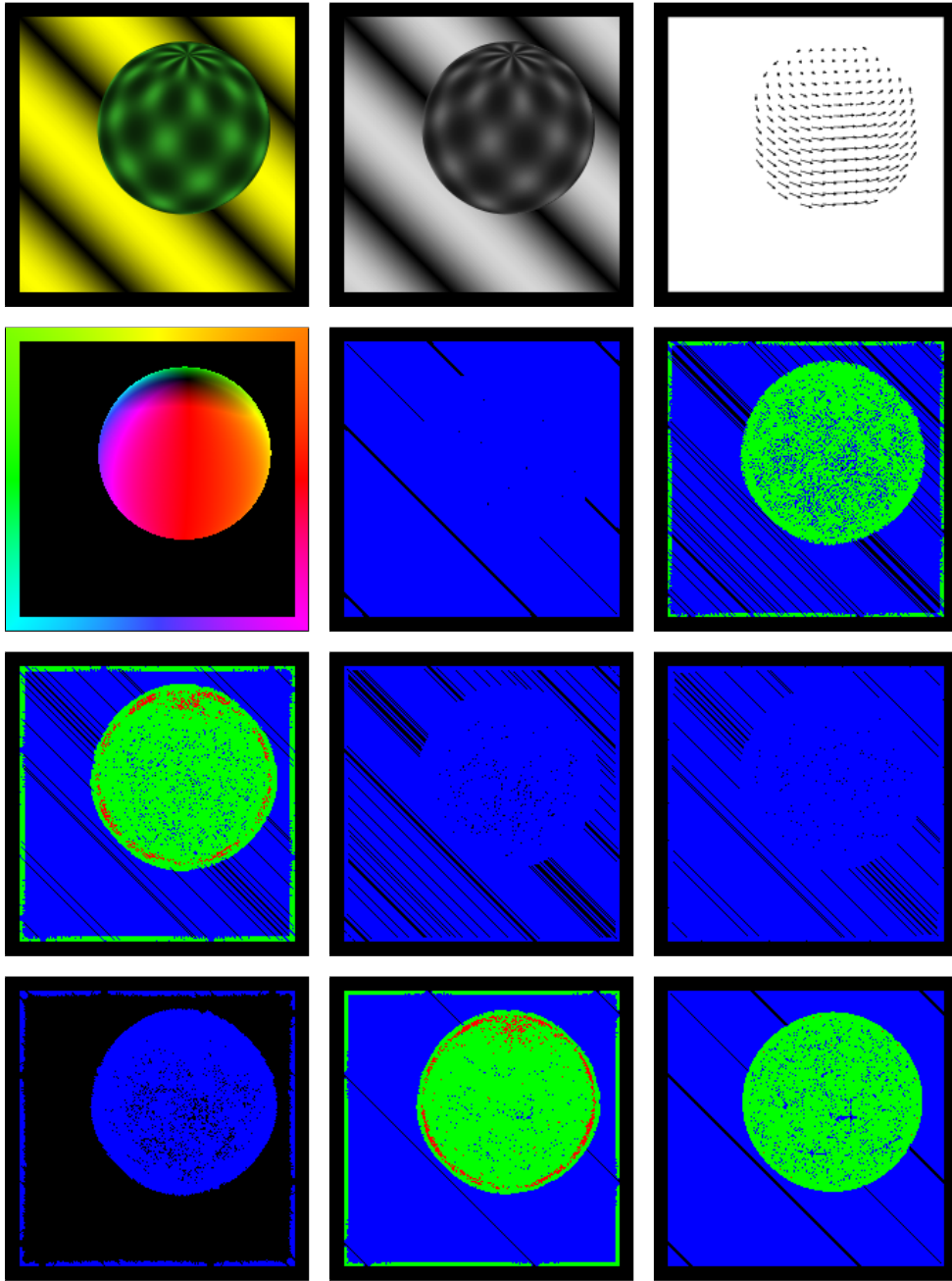
Fig. 2.6:   **Experiment I**: Impact of the constancy assumption on the *rank of the spatial motion tensor*. Visualisation of the actual rank by colour: *Black* - Rank deficiency of 2. *Blue* - Rank deficiency of 1. *Green* - Optimal Rank (no constancy assumptions violated). *Red* - Optimal Rank (constancy assumptions violated). **(a) Top Left:** Frame 10 of the *Rotating Sphere* sequence of size 200×200 pixels (colour variant). **(b) Top Centre:** Grey-value variant. **(c) Top Right:** Ground truth (vector plot). **(d) Upper Middle Left:** Ground truth (colour plot). **(e) Upper Middle Center:** Rank map for motion tensor J_1 (brightness constancy). **(f) Upper Middle Right:** Motion tensor J_2 (gradient constancy). **(g) Lower Middle Left:** Motion tensor J_3 (constancy of Hessian). **(h) Lower Middle Center:** Motion tensor J_4 (gradient magnitude constancy). **(i) Lower Middle Right:** Motion tensor J_5 (constancy of Laplacian). **(j) Bottom Left:** Motion tensor J_6 (constancy of Hessian determinant). **(k) Bottom Center:** Motion tensor J_7 (constancy of brightness and gradient). **(l) Bottom Right:** Motion tensor J_8 (constancy of colour brightness; used for colour version of the sequence).

In this context it is common to decrease the weight of neighbouring constraints with increasing distance to the center, e.g. by performing a Gaussian weighted least squares fit. This reflects the additional assumption that information obtained further away from the centre of integration is less reliable than information acquired closer to it.

R1 - Least Squares on the Brightness Constancy. Let us now apply such a Gaussian weighted least square fit to the data term for the brightness constancy assumption given by M_1. Then, the corresponding data term reads

$$M_9(D^1 f, \mathbf{u}) := \mathbf{u}^\top J_9(\nabla_3 f) \, \mathbf{u} \,, \tag{2.27}$$

where the new motion tensor J_9 is obtained from the original one J_1 via componentwise convolution with a Gaussian K_ρ of standard deviation ρ:

$$J_9(\nabla_3 f) = K_\rho * J_1(\nabla_3 f). \tag{2.28}$$

Since the standard deviation determines the area over which neighbourhood information is locally integrated, this parameter is also referred to as *integration scale* [BWS02].

In fact, for a non-zero integration scale, the proposed data term can be used as a standalone version of a local differential optic flow method. This is due to the fact that such methods are also relying on additional constraints from the neighbourhood to overcome the rank deficiency problem [LK81, BGW91, SAH91]. If, however, no integration is desired, ρ can be set to zero. Then, the proposed least square fit by minimising M_9 comes down to the original data term M_1.

In general, the strategy of local least squares fitting can also be applied to all other data terms presented so far. However, one should note that in the case of the brightness constancy assumption, the associated motion tensor is very special: It coincides exactly with the well-known *linear structure tensor* (see e.g. [BGW91, FG87, RS91]).

2.2.5.2 Adaptive Averaging with Nonlinear Diffusion

Although the preceding integration of local information by means of a Gaussian convolution is a good concept for achieving robustness under noise, the integration relies on the underlying assumption that the optic flow field is constant within the local neighbourhood described by the Gaussian kernel. Evidently, this assumption is not valid if the area of integration contains any discontinuities with respect to the flow field. As a consequence, the Gaussian convolution compromises the flow estimation in these cases and the quality of the result becomes worse. As a remedy, one can assume that the flow field is only *piecewise* constant. Then, one replaces the non-adaptive averaging scheme of the local least squares fit (2.28) that is based on Gaussian convolution – or equivalently linear diffusion – by an adaptive one [WB02, BW02] that uses nonlinear tensor-valued diffusion instead. Since nonlinear diffusion reduces the amount of smoothing at discontinuities, it avoids the integration of unrelated data beyond these motion boundaries. Consequently, it leads to less ambiguity in the least square regression.

R2 - Adaptive Averaging of the Brightness Constancy. Since the motion tensor is a matrix field, a matrix-valued scheme for nonlinear diffusion is needed. Such a scheme is proposed in [TD01] where the matrix channels are coupled by a joint diffusivity. Using the motion tensor $J_1(\nabla_3 f)$ associated to the grey value constancy assumption as initial value for the nonlinear diffusion process

$$\partial_t J_{ij} = \mathrm{div}\left(g\Big(\sum_{k,l=1}^{3} |\nabla J_{kl}|^2 \Big) \nabla J_{ij} \right) \qquad (i,j = 1, 2, 3) \tag{2.29}$$

the solution $J_1(\nabla_3 f, t)$ constitutes the computed motion tensor for a certain diffusion time t. This diffusion time is the scale parameter of the new motion tensor. It is similar to the standard deviation of the Gaussian kernel used in the least squares fit (2.28) and thus steers the size of the local neighbourhood. The so-called diffusivity function g is a decreasing function that reduces the amount of smoothing at discontinuities in the data. An appropriate choice for it is, for instance, the regularised TV diffusivity [ROF92, NS98] given by

$$g(s^2) = \frac{1}{2\sqrt{s^2 + \epsilon_{\mathrm{D}}^2}}, \tag{2.30}$$

where the small positive constant ϵ_{D} is introduced for theoretical reasons and in order to avoid unbounded diffusivities. In practice this constant can be set to a very small value, e.g. to 10^{-3}.

Let us now apply this nonlinear averaging scheme to the motion tensor of the brightness constancy assumption (M_1). Then, we obtain the data term

$$M_{10}(D^1 f, \mathbf{u}) := \mathbf{u}^\top J_{10}(\nabla_3 f)\, \mathbf{u}, \tag{2.31}$$

where the new motion tensor is obtained from the original one by the matrix-valued nonlinear diffusion process described above with diffusion time t:

$$J_{10}(\nabla_3 f) = J_1(\nabla_3 f, t). \tag{2.32}$$

As in the case of the local least squares fit, this adaptive averaging scheme can be applied to all other data terms presented so far. However, also in this case the brightness constancy assumption is somewhat special: It yields the so-called *nonlinear structure tensor* [WB02, BW02]. Alternative ways of creating adaptive tensor schemes are studied in [NG98] and [BvdBL$^+$06]. Of course, also these concepts can directly be used to create discontinuity-preserving motion tensors.

A final aspect that is worth noting is, that if one chooses the diffusivity function

$$g(s^2) = 1, \tag{2.33}$$

one ends up with homogeneous diffusion, which does not adapt to the data. Homogeneous diffusion with diffusion time t is equivalent to Gaussian convolution with standard deviation $\rho = \sqrt{2t}$. This shows the direct relation between the employment of the non-adaptive averaging in $J_9 = K_\rho * J_1$ and the adaptive one in $J_{10} = J_1(t)$.

2.2.5.3 Nonquadratic Penalisation

So far we have only considered data terms that penalise deviations from constancy assumptions in a quadratic way. From a statistical viewpoint, however, it seems desirable to penalise outliers less severely than in a quadratic setting. In particular with regard to the preservation of discontinuities in the data term, this concept from robust statistics [HRRS86, Hub81] proves to be very useful; see e.g. [BA91, HSSW02, MP98a, AK02]. In order to guarantee well-posedness for the remaining problem and allow the construction of simple globally convergent algorithms it is advantageous to use penalisers $\Psi(s^2)$ that are convex in s. Such penalisers comprise e.g. the regularised L_1 norm given by

$$\Psi_{\mathrm{D}}(s^2) = \sqrt{s^2 + \epsilon_{\mathrm{D}}^2}, \tag{2.34}$$

where ϵ_{D} is a small positive constant.

In Figure 2.7 the graphs of the corresponding functions are depicted. Apart from TV penalisation also an example for a nonconvex function is shown. Although such penalisers are more robust from a statistical viewpoint, they result in energy functionals that have multiple minima. As a consequence, minimisation strategies do usually not succeed in finding the global minimum, which in turn makes the minimisation a non-trivial task.
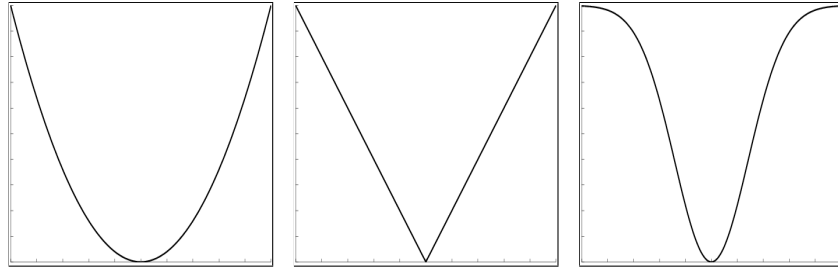
Fig. 2.7:   Comparison of different penalising functions. **(a) Left:** Tikhonov (quadratic). **(b) Center:** Total variation (linear). **(c) Right:** Example of a nonconvex function.

R3/R4 - Nonquadratic Brightness Constancy/Least Squares on Brightness Constancy.
Let us demonstrate this robustification strategy by replacing the quadratic penaliser in M_1 (brightness constancy) and M_9 (least squares on brightness constancy) by one of the proposed convex functions. Then, we obtain the following data terms:

$$M_{11}(D^1 f, \mathbf{u}) \quad := \quad \Psi_{\mathrm{D}}(\mathbf{u}^\top J_1(\nabla_3 f)\, \mathbf{u}), \tag{2.35}$$

$$M_{12}(D^1 f, \mathbf{u}) \quad := \quad \Psi_{\mathrm{D}}(\mathbf{u}^\top J_9(\nabla_3 f)\, \mathbf{u}) \tag{2.36}$$

$$= \quad \Psi_{\mathrm{D}}\Big(\mathbf{u}^\top \big(K_\rho * J_1(\nabla_3 f)\big)\, \mathbf{u}\Big).$$

Please note that the data term $M_{12}(D^1 f, \mathbf{u})$ is thereby of particular interest, since it extends the already noise robust local least square fit of CLG methods [BWS02] by an additional robust concept.

R5/R6 - Joint/Separate Robustification of the Brightness and Gradient Constancy. In the case of a single constraint, the realisation of the non-quadratic penalisation is straightforward. If the data term, however, is based on multiple constraints, the robustification becomes more complicated. In particular the problem occurs how the different terms shall be penalised. In this context, two different strategies that can be used:

(i) *Joint Robustification.* The basic idea of the joint robustification strategy is to consider the whole data term as one entity that represents all constancy assumptions that are imposed on the image data. Consequently, all assumptions are robustified jointly – by applying a single robust function to the complete data term [BBPW04].

(ii) *Separate Robustification.* In contrast to the joint concept, the separate robustification strategy considers all constancy assumptions as *independent* constraints that can be fulfilled or violated separately. From this point of view it obviously makes more sense to robustify each of the constancy assumptions individually [BW05].

Let us illustrate both concepts by the example of the combined data term M_7 (brightness and gradient constancy). While the data term based on a joint robustification is given by

$$
\begin{aligned}
M_{13}(D^1 f, D^2 f, \mathbf{u}) \quad &:= \quad \Psi_{\mathrm{D}}(\mathbf{u}^\top J_7(\nabla_3 f)\,\mathbf{u}) \qquad\qquad\qquad (2.37)\\
&= \quad \Psi_{\mathrm{D}}\Big(\mathbf{u}^\top(\gamma_1\, J_1(\nabla_3 f) + \gamma_2\, J_2(\nabla_3 f))\,\mathbf{u}\Big)\\
&= \quad \Psi_{\mathrm{D}}\Big(\gamma_1\,(\mathbf{u}^\top\nabla_3 f)^2 + \gamma_2\,\sum_{i=1}^{2}(\mathbf{u}^\top\nabla_3 f_{x_i})^2\Big),
\end{aligned}
$$

the one based on the separate strategy reads

$$
\begin{aligned}
M_{14}(D^1 f, D^2 f, \mathbf{u}) \quad &:= \quad \gamma_1\,\Psi_{\mathrm{D}_1}(\mathbf{u}^\top J_1(\nabla_3 f)\,\mathbf{u}) + \gamma_2\,\Psi_{\mathrm{D}_2}(\mathbf{u}^\top J_2(\nabla_3 f)\,\mathbf{u}) \quad (2.38)\\
&= \quad \gamma_1\,\Psi_{\mathrm{D}_1}\Big((\mathbf{u}^\top\nabla_3 f)^2\Big) + \gamma_2\,\Psi_{\mathrm{D}_2}\Big(\sum_{i=1}^{2}(\mathbf{u}^\top\nabla_3 f_{x_i})^2\Big).
\end{aligned}
$$

Since in the case of the separate robustification two non-quadratic functions are required, this gives us one additional degree of freedom. However, if both constancy assumptions can be expected to give the same types of outliers – this is obviously the case for the brightness and the gradient constancy assumption – they should also be treated equally. Therefore it makes sense to use the same function $\Psi_{\mathrm{D}} = \Psi_{\mathrm{D}_1} = \Psi_{\mathrm{D}_2}$ to penalise deviations from both constancy assumptions.

One should note that unlike the quadratic data terms M_9 and M_{10}, the non-quadratic ones M_{11}-M_{14} have no motion tensor equivalent. This is due to the fact that, evidently, they cannot be written as a quadratic form. However, the motion tensor notation may still be useful: As part of the quadratic argument of the non-quadratic penaliser, the motion tensor still allows to decide, whether the rank deficiency problem is locally present or not.

An overview of all robust data terms presented so far (M_9–M_{14}) and their capability of handling discontinuities in the data is given in Table 2.5.

Tab. 2.5: Comparison of the robust data terms M_9–M_{14}.

	data term	concept	discontinuities
M_9	$\mathbf{u}^\top\Big(K_\rho * J_1(\nabla_3 f)\Big)\,\mathbf{u}$	least squares	no
M_{10}	$\mathbf{u}^\top J_1(\nabla_3 f, t)\,\mathbf{u}$	nonlinear diffusion	yes
M_{11}	$\Psi\Big(\mathbf{u}^\top J_1(\nabla_3 f)\mathbf{u}^\top)\Big)$	nonquadratic penaliser	yes
M_{12}	$\Psi\Big(\mathbf{u}^\top\Big(K_\rho * J_1(\nabla_3 f)\Big)\,\mathbf{u}\Big)$	least squares + nonquadratic penaliser	yes
M_{13}	$\Psi_\mathrm{D}\Big(\mathbf{u}^\top\Big(\gamma_1\,J_1(\nabla_3 f) + \gamma_2\,J_2(\nabla_3 f)\Big)\,\mathbf{u}\Big)$	nonquadratic penaliser (joint robustification)	yes
M_{14}	$\gamma_1\,\Psi_{\mathrm{D}_1}\Big(\mathbf{u}^\top J_1(\nabla_3 f)\,\mathbf{u}\Big)$ $+$ $\gamma_2\,\Psi_{\mathrm{D}_2}\Big(\mathbf{u}^\top J_2(\nabla_3 f)\,\mathbf{u}\Big)$	nonquadratic penaliser (separate robustification)	yes

2.2.6 Robust Data Terms - Experiments

After we have discussed different strategies for rendering the data term more robust with respect to noise and outliers in the image data, let us now investigate their impact on the quality of the computed flow field. This shall be done in the following two experiments.

2.2.6.1 Experiment I: Gaussian Noise

In our first experiment we compare different data terms regarding their robustness under noise. To this end, we have added Gaussian noise with zero mean and varying standard deviation σ_n to the Yosemite sequence *with* clouds. Apart from the data term M_1 (brightness constancy), we considered its robust variants M_9 and M_{10} that are both based on the concept of local integration (least square fit and nonlinear diffusion, respectively). Moreover, we chose with M_2 (gradient constancy) a data term, that shall represent the behavior of constancy assumptions that are based on derivatives of the image sequence. The obtained results for the different data terms and noise levels are presented in Table 2.6. They show the following two facts:

- *Usefulness of Local Integration under Noise.* Comparing the results of M_1, M_9 and M_{10}, one can see that both robust data terms give better results than their non-robust counterpart. This can be explained by the additional neighbourhood information that is used when locally integrating the motion tensor. One should note that the variant based on nonlinear diffusion is better for a small amount of noise while the least square fit seems more suitable for noise of a larger standard deviation. This is due to the fact that nonlinear diffusion *adapts* to the image data and thus allows to preserve discontinuities. While this adaptation improves the results for a small

amount of noise, it may lead to an adaptation to noise itself if σ_n becomes larger. Of course, this constitutes no problem for the local least square fit: In contrast to nonlinear diffusion, it is based on a *fixed* scale of integration.

- *Increased Sensitivity of Higher Order Derivatives with Respect to Noise.* A second aspect that can be observed, is the poor performance of the data term M_2 (gradient constancy). Although the underlying assumption is one of the most suitable ones for the Yosemite sequence *with* clouds (due to the varying illumination in the sky), the corresponding data term is outperformed significantly for large amounts of noise - even by the data term that makes use of the less suitable brightness constancy (M_1). This, however, is not surprising: The computation of higher order derivatives is well-known to be noise sensitive. Of course, this does not constitute a real problem: Also the gradient constancy assumption can be robustified.

The computed flow fields for the best performing data term M_9 (least squares) are depicted in Figure 2.8. Although the original sequence was degraded severely (in particular for $\sigma_n = 40$), the obtained results still look realistic. However, one notices that the flow field becomes smoother with increasing noise level. This is a consequence of the large values for all three smoothness parameters – the noise scale σ, the smoothness weight α and the integration scale ρ – that are required to deal with the increasing perturbations (cf. Table 2.6). For a more detailed discussion on the different roles of this three parameters the reader is referred to [BWS05].

Tab. 2.6: **Experiment I**: Performance of robust data terms under *Gaussian noise*. Comparison of data terms M_1, M_2, M_9 and M_{10}. We added Gaussian noise with varying standard deviations σ_n to the Yosemite sequence with clouds and used a spatial energy functional with homogeneous regularisation to compute the average angular error (AAE). The parameters σ, α, ρ, and t have been optimised.

noise	data term	σ	α	integration parameter	AAE
$\sigma_n = 0$	M_1	1.30	500	-	7.17°
	M_9	1.30	500	$\rho = 1.80$	7.14°
	M_{10}	1.30	300	$t = 250$	6.97°
	M_2	2.10	20	-	5.91°
$\sigma_n = 10$	M_1	1.77	1100	-	9.37°
	M_9	1.77	950	$\rho = 4.50$	9.18°
	M_{10}	1.90	850	$t = 200$	9.02°
	M_2	3.00	27	-	8.88°
$\sigma_n = 20$	M_1	2.08	2200	-	12.17°
	M_9	2.09	1600	$\rho = 10.70$	11.71°
	M_{10}	2.10	1600	$t = 225$	11.76°
	M_2	3.60	35	-	12.26°
$\sigma_n = 40$	M_1	2.45	4100	-	16.80°
	M_9	2.38	2000	$\rho = 17.60$	15.82°
	M_{10}	2.40	2500	$t = 500$	16.29°
	M_2	4.20	55	-	18.00°
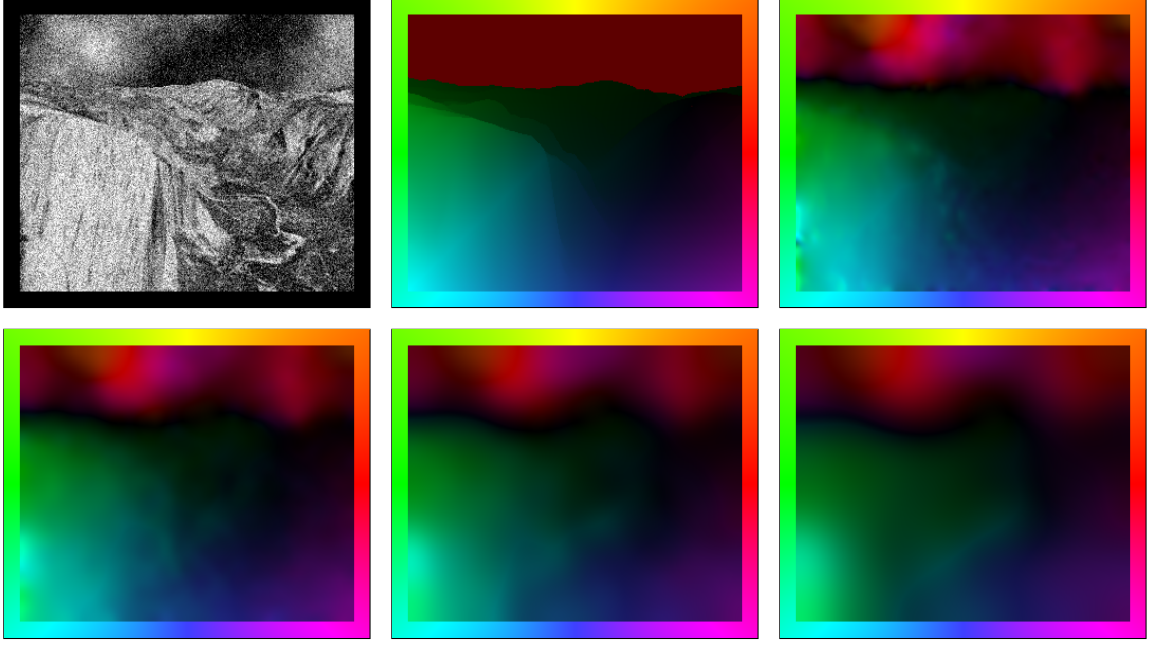
Fig. 2.8:   **Experiment I**: Performance of robust data terms under *Gaussian noise*. Results for the data term M_9 (least squares on brightness constancy). **(a) Top Left:** Frame 8 of the *Yosemite* sequence *with* clouds degraded by Gaussian noise of standard deviation $\sigma_n^2 = 0$. **(b) Top Centre:** Ground truth (colour plot). **(c) Top Right:** Computed flow field for a spatial approach with data term M_9 (least squares on brightness constancy) and homogeneous regularisation. Noise level $\sigma_n = 0$. **(d) Bottom Left:** Noise level $\sigma_n = 10$. **(e) Bottom Middle:** Noise level $\sigma_n = 20$. **(f) Bottom Right:** Noise level $\sigma_n = 40$.

2.2.6.2   Experiment II: Outliers in the Data

In our second experiment we investigate the advantages of nonquadratic penalisers. This is done in Table 2.7, where the quadratic data terms M_1 (brightness constancy) and M_7 (brightness and gradient constancy) are compared to their nonquadratic counterparts M_{11} (nonquadratic penaliser) and M_{12} (nonquadratic penaliser and least squares) and M_{13} (nonquadratic penaliser, joint robustification) and M_{14} (nonquadratic penaliser, separate robustification), respectively. Again, the listed results refer to the Yosemite sequence *with* clouds. Here, the following two observations can be made:

- *Usefulness of Nonquadratic Penalisers with Respect to Outliers.* Obviously, one can improve the estimation by replacing the quadratic penaliser with a nonquadratic one. This is due to the fact that in the case of divergent motion – such a motion pattern occurs in the Yosemite sequence – boundary pixels from the first frame are not necessarily present in the second one. Evidently, these pixels – that do not have a corresponding counterpart in the second frame – are violating any constancy assumption per definition. By penalising outliers in a non-quadratic way, the influence of these pixels on the overall result is reduced significantly. As a consequence, the estimation quality improves (cf. average angular error). In this context we like to point out that a comparison between the results of the variants for M_1 and M_7 would not be fair. The significantly lower average angular errors of the data terms originat-

ing from M_7 are due to the gradient constancy assumption that allows to overcome the varying illumination in the sky.

- *Usefulness of Separate Robustification.* As the achieved average angular error shows, the separate robustification gives slightly better results than its joint counterpart. This can be explained as follows: When replacing the joint robustification of two constraints s_1 and s_2 by a separate one using the L_1 norm, the new data term penalises exactly those solutions more severely, where both constraints do not hold, i.e. $|s_1| \neq 0$ and $|s_2| \neq 0$. This can be easily verified by comparing both variants, namely

$$\gamma_1 \Psi_\mathrm{D}(s_1^2) + \gamma_2 \Psi_\mathrm{D}(s_2^2) = \gamma_1|s_1| + \gamma_2|s_2| = \sqrt{\gamma_1^2 s_1^2 + 2\gamma_1\gamma_2|s_1||s_2| + \gamma_2^2\, s_2^2}$$

and

$$\Psi_\mathrm{D}(\gamma_1^* s_1^2 + \gamma_2^* s_2^2) = \sqrt{\gamma_1^* s_1^2 + \gamma_2^* s_2^2}.$$

while setting $\gamma_1 = \sqrt{\gamma_1^*}$ and $\gamma_2 = \sqrt{\gamma_2^*}$. One should note that this property also holds for other robust functions as long as they are positive and concave in s^2. Such functions lead in the separate case always to a stronger penalisation of those results that are considered as outliers in both constancy assumptions. This in turn favours exactly those solutions where at least one of the constancy assumptions holds – and thus reflects the assumption that they can hold *independently* from each other.

The corresponding flow fields given in Figure 2.9 confirm our considerations. In particular, in the lower left corner - this region is shown once again in Figure 2.10 – the advantages of robust penalisers become obvious. In this area the problem of outliers is present: A border of several pixels is moved out of the image from one frame to the next. In the case of the quadratic data terms, these outliers cause severe artifacts that are clearly visible. In the case of the nonquadratic data terms, however, these artifacts disappear almost completely. In fact, the resulting flow field is very smooth, which coincides much better with the actual motion.

Tab. 2.7: **Experiment II**: Performance of robust data terms with respect to *outliers in the data*. Comparison of quadratic and nonquadratic penalisers for the data terms M_1 (brightness constancy) and M_7 (brightness and gradient constancy). We used a spatial energy functional with homogeneous regularisation, and computed the average angular error (AAE) for the Yosemite sequence with clouds. The parameters σ, α and ρ have been optimised.

data term	penaliser	σ	α	other parameters	AAE
M_1	quadratic	1.30	500	-	7.17°
M_{11}	nonquadratic	1.40	190	-	7.08°
M_{12}	nonquadratic + least squares	1.40	200	ρ=2.0	6.76°
M_7	quadratic	2.10	21	γ_1=0.01, γ_2=1	5.87°
M_{13}	nonquadratic (joint robustification)	2.10	400	γ_1=0.06, γ_2=100	5.18°
M_{14}	nonquadratic (separate robustification)	2.10	400	γ_1=0.06, γ_2=10	5.10°
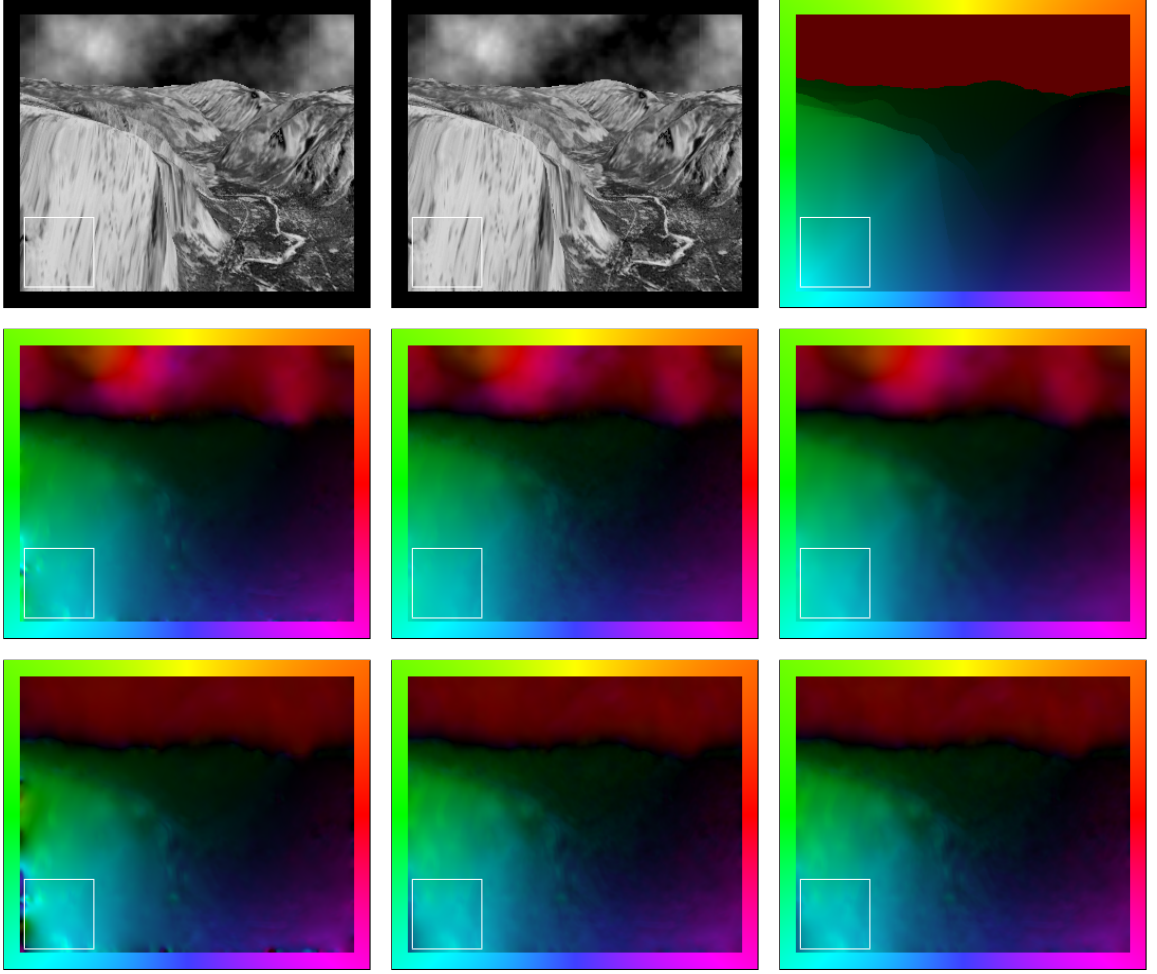
Fig. 2.9: **Experiment II**: Performance of robust data terms with respect to *outliers in the data* (1/2). Comparison of quadratic and nonquadratic penalisers for the data terms M_1 (brightness constancy) and M_7 (brightness and gradient constancy). **(a) Top Left:** Boundary detail from Frame 8 of the Yosemite sequence *with* clouds (64×64 pixels). **(b) Top Centre:** Frame 9. **(c) Top Right:** Ground truth. **(d) Middle Left:** Computed flow field for a spatial approach with data term M_1 (quadratic penaliser) and homogeneous regularisation. **(e) Middle Centre:** Data term M_{11} (nonquadratic penaliser). **(f) Middle Right:** Data term M_{12} (nonquadratic penaliser and least squares). **(g) Bottom Left:** Data term M_7 (quadratic penaliser). **(h) Bottom Centre:** Data term M_{13} (nonquadratic penaliser, joint robustification). **(i) Bottom Right:** Data term M_{14} (nonquadratic penaliser, separate robustification).

2.3 Smoothness Terms

In our section on data terms we have shown how the integration of prior knowledge on the scene can influence the quality of the estimation. However, we have also seen that the local data alone may not be sufficient to compute a locally unique solution. In order to overcome this problem, variational optic flow methods make use of an additional term, that incorporates prior knowledge on the solution itself: *the smoothness term*. In this section that it based on the article in [WBBP06], we discuss which smoothness assumptions are commonly used in the literature and how these assumptions can be classified in a suitable way. To this end, we follow the idea of Weickert and Schnörr [WS01a], and link the design
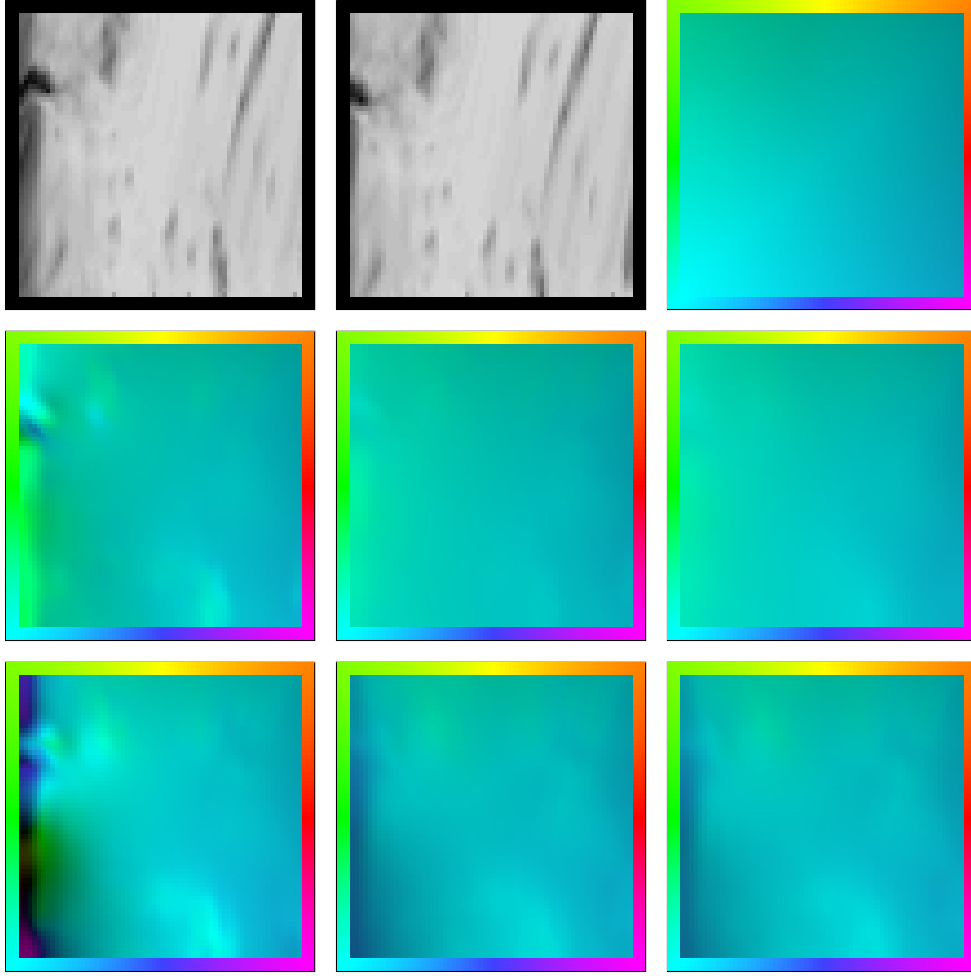
Fig. 2.10: **Experiment II**: Performance of robust data terms with respect to *outliers in the data* (2/2). Comparison of quadratic and nonquadratic penalisers for the data terms M_1 (brightness constancy) and M_7 (brightness and gradient constancy). **(a) Top Left:** Boundary detail from Frame 8 of the Yosemite sequence *with* clouds (64×64 pixels). **(b) Top Centre:** Frame 9. **(c) Top Right:** Ground truth. **(d) Middle Left:** Computed flow field for a spatial approach with data term M_1 (quadratic penaliser) and homogeneous regularisation. **(e) Middle Centre:** Data term M_{11} (nonquadratic penaliser). **(f) Middle Right:** Data term M_{12} (nonquadratic penaliser and least squares). **(g) Bottom Left:** Data term M_7 (quadratic penaliser). **(h) Bottom Centre:** Data term M_{13} (nonquadratic penaliser, joint robustification). **(i) Bottom Right:** Data term M_{14} (nonquadratic penaliser, separate robustification).

of smoothness terms to the modelling of multi-channel diffusion processes. Furthermore, we investigate the usefulness of spatiotemporal smoothness constraints. In contrast to their spatial counterparts, such constraints consider a temporal window of more than two frames for the estimation of the flow field.

2.3.1 A Diffusion Based Classification of Smoothness Terms

As shown by Weickert and Schnörr in [WS01a] the design of smoothness terms is closely related to the desired type of *filling-in* effect. In fact, the smoothness term results in a diffusion process that takes information from the neighbourhood and propagates it to lo-

cations, where the rank deficiency problem is present, i.e. where no unique solution of the data term can be computed. This connection between regularisation methods and diffusion filters becomes evident, when minimising the underlying energy functional via its *steepest descent equations*: The resulting expressions constitute a *diffusion–reaction system*, where the obtained diffusion part depends only on the original smoothness term. Weickert and Schnörr used this implicit relation to set up a diffusion based taxonomy for smoothness terms. The different smoothness constraints were thereby classified in accordance with the induced diffusion process [Wei98]. In order to develop a deeper understanding of this diffusion based taxonomy, let us start by analysing how the underlying diffusion–reaction system is obtained.

2.3.1.1   From Energy Functionals to Diffusion–Reaction Systems

Minimising the energy functional (1.3) can be done in two different ways: Either one performs an *elliptic approach* by solving the corresponding Euler–Lagrange equations, or one performs a *parabolic approach*, where one is interested in finding the steady state of the corresponding diffusion–reaction system. In the following both strategies are discussed in detail.

The Elliptic Approach. One possibility for minimising an energy functional is the computation of its so-called *Euler–Lagrange equations*. The Calculus of Variations tells us that these equations constitute a necessary condition for each minimiser of $E(u)$ [CH53, Els61, GF00]. In the specific case of a spatial energy functional (1.3) they are given by the following two-dimensional system of partial differential equations (PDEs)

$$0 \;=\; \partial_{x_1} S_{u_{1x_1}} + \partial_{x_2} S_{u_{1,x_2}} - \tfrac{1}{\alpha}\; \partial_{u_1} M, \qquad (2.39)$$

$$0 \;=\; \underbrace{\partial_{x_1} S_{u_{2x_1}} + \partial_{x_2} S_{u_{2,x_2}}}_{\text{smoothness}} - \tfrac{1}{\alpha}\; \underbrace{\partial_{u_2} M}_{\text{data}} \qquad (2.40)$$

equipped with homogeneous Neumann (reflecting) boundary conditions. Here, the term $S_{u_{i,x_j}}$ denotes the partial derivative of S with respect to $\partial_{x_j} u_i$.

The Parabolic Approach. Another possibility for minimising $E(u)$ is the use of the *steepest descent method*. In the case of a spatial functional this yields a system of two-dimensional diffusion–reaction equations, where the diffusion term results from the regulariser $S(\nabla f, \nabla u)$, and the reaction term is induced by the data term $M(D^k f, u)$:

$$\partial_t u_1 \;=\; \partial_{x_1} S_{u_{1x_1}} + \partial_{x_2} S_{u_{1,x_2}} - \tfrac{1}{\alpha}\; \partial_{u_1} M, \qquad (2.41)$$

$$\partial_t u_2 \;=\; \underbrace{\partial_{x_1} S_{u_{2x_1}} + \partial_{x_2} S_{u_{2,x_2}}}_{\text{diffusion}} - \tfrac{1}{\alpha}\; \underbrace{\partial_{u_2} M}_{\text{reaction}} \qquad (2.42)$$

Here, the parameter t is a pure numerical parameter that denotes the evolution time. It should not be confused with the time x_3 of the image sequence. In order to find a minimum of the underlying energy functional, one has to compute the steady–state of this evolution equation, i.e. the solution for $t \to \infty$. It is evident that this solution also fulfills the Euler–Lagrange equations (2.39)–(2.40), since in this case $\partial_t u_1 = 0$ and $\partial_t u_2 = 0$. If

$E(u)$ is *strictly convex* it even constitutes a *unique minimiser*. Moreover, in this case, the steepest descent evolution is *globally convergent*: Its steady–state does not depend on the initialisation.

Since we are interested in a taxonomy for optic flow regularisers, it is sufficient to restrict ourselves to the diffusion part of (2.41)–(2.42). This diffusion part can be identified with a *multi-channel* diffusion process on the two channels u_1 and u_2:

$$\partial_t u_i \;=\; \partial_{x_i} S_{u_{i,x_1}} + \partial_{x_i} S_{u_{i,x_2}} \qquad (i = 1, 2). \tag{2.43}$$

In order to get a better understanding of such processes, it is instructive to make a little excursion to diffusion filters for multi-channel images. This shall be done next.

2.3.1.2   Diffusion of Multi-Channel Images - Modelling

In our section on data terms we have already addressed colour images as a special form of multi-channel image data. However, there are also other typical representatives such as satellite images and multi-spin echo MR data sets. They all have in common that they require special filtering schemes that consider a joint treatment of all channels. Such schemes do also exist for diffusion filters: Let $\mathbf{f} = (f_1(\mathbf{x}), ..., f_m(\mathbf{x}))^\top$ be some multi-channel image with $\mathbf{x} \in \mathbb{R}^2$. Then, a suitable scheme for a diffusion filter is described by the following evolution

$$\partial_t u_i \;=\; \operatorname{div} \Big(\, T(\nabla \mathbf{f}, \nabla \mathbf{u}) \, \nabla u_i \Big) \qquad (i = 1, ..., m) \tag{2.44}$$

where $f(x)$ acts as initial condition for the solution $\mathbf{u}(x, t)$:

$$u_i(\mathbf{x}, 0) = f_i(\mathbf{x}) \qquad (i = 1, ..., m). \tag{2.45}$$

Here, $T(\nabla \mathbf{f}, \nabla \mathbf{u})$ denotes the so-called *diffusion tensor* – a *positive definite* 2×2 matrix that steers the direction and intensity of the local diffusion process (jointly for all channels). Often this steering is adapted such that it respects discontinuities either in the initial image $\mathbf{f}$ or the evolving image $\mathbf{u}$. Depending on the form of the diffusion tensor, five different cases can be distinguished: *homogeneous* diffusion, *linear isotropic* and *linear anisotropic diffusion* as well as *nonlinear isotropic* and *nonlinear anisotropic* diffusion.

Before we will discuss this five diffusion types, let us point out that the corresponding models are not the only PDE methods that have been proposed for processing multi-channel images. For alternative approaches the reader is referred to [BC98, KMS00, Sap01, Wei99] and the references therein. However, in contrast to other approaches, our taxonomy allows a connection to optic flow regularisers: It is based on diffusion processes in divergence form that can be derived as steepest descent methods for minimising suitable energy functionals. Let us start our classification with the most popular diffusion process: Homogeneous diffusion.

T1 - Homogeneous Diffusion. Homogeneous diffusion is the simplest form of diffusion. The corresponding diffusion tensor comes down to the identity matrix and is given by

$$T_1(\nabla \mathbf{f}, \nabla \mathbf{u}) = I. \tag{2.46}$$

It describes a diffusion process that treats all directions equally and does neither adapt to discontinuities in the initial nor in the evolving image [Iij59]. As a consequence, the

obtained image becomes homogeneously blurred and semantically important edges may get dislocated.

T2 - Linear Isotropic Diffusion. A more advanced diffusion process is obtained if the diffusion tensor is modified in such a way that it respects discontinuities in the *initial* image $\mathbf{f}$; see [Fri92]. This requires two properties: (i) The diffusion tensor must be able to detect discontinuities in $\mathbf{f}$. This can be realised by evaluating the *joint* gradient of all image channels $\sum_{j=1}^{m} |\nabla f_j|^2$. (ii) The diffusion tensor must inhibit the diffusion process if this gradient is large, i.e. discontinuities are present. This in turn can be achieved by using a positive function $g(s^2)$ that decreases in s^2 and is close to zero for $s^2 \to \infty$. Such a function is for instance given by the regularised TV diffusivity (2.30). Performing both modifications with respect to T_1 the following diffusion tensor is obtained:

$$T_2(\nabla \mathbf{f}, \nabla \mathbf{u}) = g\Big(\sum_{j=1}^{m} |\nabla f_j|^2 \Big) I. \tag{2.47}$$

One should note that the corresponding diffusion process is called *linear*, since T_2 only depends on the initial image (which does not change over time). Moreover, the process is said to be *isotropic*, since all directions are treated equally (although the strength of the diffusion may vary).

T3 - Linear Anisotropic Diffusion. Until now, we have restricted ourselves to diffusion processes that are isotropic. However, in particular with respect to discontinuities, it makes sense to distinguish between different directions [Iij62]: While diffusion *across* discontinuities is not desired, diffusion *along* edges may even improve the filtering result – in particular with respect to noisy boundaries. Such an *anisotropic* behaviour ca be achieved by extracting structural information from the joint image tensor $\sum_{j=1}^{m} \nabla f_j \nabla f_j^\top$ by means of an eigenvalue decomposition:

$$\sum_{j=1}^{m} \nabla f_j \nabla f_j^\top = (\mathbf{v}_1, \mathbf{v}_2) \left(\begin{array}{cc} \lambda_1 & 0 \\ 0 & \lambda_2 \end{array} \right) (\mathbf{v}_1, \mathbf{v}_2)^\top. \tag{2.48}$$

While the eigenvectors $\mathbf{v}_1, \mathbf{v}_2$ represent the directions with highest and lowest contrast, the corresponding eigenvalues λ_1, λ_2 specify the value for the contrast itself. A suitable diffusion tensor can then be created by keeping the original eigenvectors fixed and modifying the eigenvalues by applying a function $g(s^2)$ as in the isotropic case:

$$G \left(\sum_{j=1}^{m} \nabla f_j \nabla f_j^\top \right) = (\mathbf{v}_1, \mathbf{v}_2) \left(\begin{array}{cc} g(\lambda_1) & 0 \\ 0 & g(\lambda_2) \end{array} \right) (\mathbf{v}_1, \mathbf{v}_2)^\top. \tag{2.49}$$

This strategy, that inhibits diffusion across dominant features of the image, allows the extension of any function g to the tensor-valued domain (and codomain). In the following we will abbreviate the associated diffusion tensor by

$$T_3(\nabla \mathbf{f}, \nabla \mathbf{u}) = G\Big(\sum_{j=1}^{m} \nabla f_j \nabla f_j^\top \Big). \tag{2.50}$$

As in the two previous cases the corresponding diffusion process is *linear*. Although more (directional) information is used, it still depends on the initial image $\mathbf{f}$.

Tab. 2.8: Comparison of the diffusion tensors T_1–T_5.

	diffusion tensor	diffusion process	discontinuities	adaptation
T_1	I	homogeneous	no	-
T_2	$g\Big(\sum\limits_{j=1}^{m} \lvert\nabla f_j\rvert^2\Big)I$	linear isotropic	yes	initial image
T_3	$G\Big(\sum\limits_{j=1}^{m} \nabla f_j \nabla f_j^\top\Big)$	linear anisotropic	yes	initial image
T_4	$g\Big(\sum\limits_{j=1}^{m} \lvert\nabla u_j\rvert^2\Big)I$	nonlinear isotropic	yes	evolving image
T_5	$G\Big(\sum\limits_{j=1}^{m} \nabla u_j \nabla u_j^\top\Big)$	nonlinear anisotropic	yes	evolving image

T4 - Nonlinear Isotropic Diffusion. So far, the discussed diffusion processes were designed in such a way that they respect discontinuities in the *initial* image. However, from a modelling point of view it may also be interesting to have a feedback from the *evolution* itself. In the isotropic case, this can be achieved by replacing the joint gradient of the initial image $\sum_{j=1}^{m} \lvert\nabla f_j\rvert^2$ by the joint gradient of the evolving image $\sum_{j=1}^{m} \lvert\nabla u_j\rvert^2$. Then, one obtains the diffusion tensor [GKKJ92]

$$T_4(\nabla\,\mathbf{f}, \nabla\,\mathbf{u}) = g\Big(\sum_{j=1}^{m} \lvert\nabla u_j\rvert^2\Big)I, \tag{2.51}$$

where $g(s^2)$ is a positive decreasing function as defined before. One should note that this allows the underlying diffusion process e to adapt to discontinuities that are created *during* the evolution. Evidently, this *nonlinear* behaviour is a significant different form the one of linear diffusion: The corresponding diffusion tensors do not change over time.

T5 - Nonlinear Anisotropic Diffusion Of course, this feedback strategy can also be applied in the anisotropic case. To this end, one has to replace the joint image tensor $\sum_{j=1}^{m} \nabla f_j \nabla f_j^\top$ by its evolution dependent counterpart $\sum_{j=1}^{m} \nabla u_j \nabla u_j^\top$. The obtained diffusion tensor is then given by [Wei94b]

$$T_5(\nabla\,\mathbf{f}, \nabla\,\mathbf{u}) = G\Big(\sum_{j=1}^{m} \nabla u_j \nabla u_j^\top\Big), \tag{2.52}$$

where $G(T)$ has to be understood once more as a function $g(s^2)$ that modifies the eigenvalues of its tensor-valued argument T. As in the previous case, the underlying diffusion process is *nonlinear*.

A summary of all presented diffusion processes in form of the corresponding diffusion tensors is given in Table 2.8.

Fig. 2.11: Diffusion filtering of colour images. **(a) Top Left:** Noisy colour image. **(b) Top Centre:** Homogeneous diffusion. **(c) Top Right:** Linear isotropic diffusion. **(d) Bottom Left:** Linear anisotropic diffusion. **(e) Bottom Centre:** Nonlinear isotropic diffusion. **(f) Bottom Right:** Nonlinear anisotropic diffusion.

2.3.1.3 Diffusion of Multi-Channel Images - Experiments

In our experiment we compare the effect of the different smoothing strategies for a noisy RGB colour image. To to this end, we applied Gaussian noise of standard deviation $\sigma_n = 40$. Figure 2.11 shows the obtained results. We observe that homogeneous diffusion performs well with respect to denoising, but does not respect image edges. Space-variant linear isotropic diffusion, however, may suffer from noise sensitivity as strong noise may be misinterpreted as an important edge structure where the diffusivity is reduced. Anisotropic linear diffusion allows smoothing along edges, but reduces smoothing across them. This leads to a better performance than isotropic linear diffusion if images are noisy. We can also observe that nonlinear models give better results than their linear counterparts. This is not surprising, since the nonlinear models adapt the diffusion process to the evolving image instead of the initial one.

Tab. 2.9: Optic flow regularisers and their corresponding vector-valued diffusion processes. In the diffusion context, f denotes the vector-valued initial image and u its evolution. In the optic flow setting, f is the scalar-valued image sequence and u describes the optic flow field.

		optic flow regulariser $S(\nabla f, \nabla u)$		vector-valued diffusion process $\partial_{x_1} S_{u_{ix_1}} + \partial_{x_2} S_{u_{ix_2}}$		
S_1	homo-geneous	$\sum\limits_{i=1}^{2} |\nabla u_i|^2$ (Horn/Schunck [HS81])		$\operatorname{div}(I \nabla u_i)$ (scalar case: Iijima [Iij59])	homo-geneous	T_1
S_2	image-driven isotropic	$g(|\nabla f|^2) \sum\limits_{i=1}^{2} |\nabla u_i|^2$ (Alvarez et al. [AELS99])		$\operatorname{div}\left(g(\sum\limits_{j} |\nabla f_j|^2)\nabla u_i\right)$ (scalar case: Fritsch [Fri92])	linear isotropic	T_2
S_3	image-driven anisotropic	$\sum\limits_{i=1}^{2} \nabla u_i^\top D(\nabla f)\, \nabla u_i$ (Nagel [Nag83a])		$\operatorname{div}\left(G(\sum\limits_{j} \nabla f_j \nabla f_j^\top)\nabla u_i\right)$ (scalar case: Iijima [Iij62])	linear anisotropic	T_3
S_4	flow-driven isotropic	$\Psi\left(\sum\limits_{i=1}^{2} |\nabla u_i|^2\right)$ (Schnörr [Sch94b])		$\operatorname{div}\left(\Psi'(\sum\limits_{j} |\nabla u_j|^2)\nabla u_i\right)$ (Gerig et al. [GKKJ92])	nonlinear isotropic	T_4
S_5	flow-driven anisotropic	$\operatorname{tr}\Psi\left(\sum\limits_{i=1}^{2} \nabla u_i \nabla u_i^\top\right)$ (Weickert/Schnörr [WS01a])		$\operatorname{div}\left(\Psi'\left(\sum\limits_{j} \nabla u_j \nabla u_j^\top\right)\nabla u_i\right)$ (Weickert [Wei94b])	nonlinear isotropic	T_5

2.3.2   Spatial Regularisation - Modelling

After we have discussed a suitable classification for multi-channel diffusion processes, we can transfer it to the optic flow setting. The idea is to identify the optic flow regularisers $S(\nabla f, \nabla u)$ that produce homogeneous, linear isotropic, linear anisotropic, nonlinear isotropic, and nonlinear anisotropic diffusion. It should be noted that now that we returned to the optic flow setting, f denotes the image sequence again, and u is the flow field.

The simplest optic flow regulariser is the *homogeneous* regularisation of Horn and Schunck [HS81]. This quadratic regulariser of type $S(\nabla u) = |\nabla u_1|^2 + |\nabla u_2|^2$ penalises all deviations from smoothness of the flow field. It can be related to linear diffusion with a constant diffusivity. Thus, the flow field is blurred in a homogeneous way such that motion discontinuities may loose sharpness and get dislocated. It is thus not surprising that people have tried to construct a variety of discontinuity-preserving regularisers. Depending on the structure of the resulting diffusion term, we can classify a regulariser $S(\nabla f, \nabla u)$

as image-driven or flow-driven, and isotropic or anisotropic [WS01a].

For *image-driven* regularisers, S is not only a function of the flow gradient ∇u but also of the image gradient ∇f. This function is chosen in such a way that it respects discontinuities in the image data. If only the gradient *magnitude* $|\nabla f|$ matters, the method is called *isotropic* [AELS99]. It can avoid smoothing at image edges. An *anisotropic* technique depends also on the *direction* of ∇f. Typically it reduces smoothing across edges of f (i.e. along ∇f), while smoothing along edges of f is still permitted [Nag83a]. Image-driven regularisers can be related to linear diffusion processes.

Flow-driven regularisers take into account discontinuities of the unknown flow field u by preventing smoothing at or across flow discontinuities. If the resulting diffusion process uses a scalar-valued diffusivity that only depends on $|\nabla u|^2 := |\nabla u_1|^2 + |\nabla u_2|^2$, it is an *isotropic* process [Sch94b]. Cases where also the direction of ∇u_1 and ∇u_2 matters are named *anisotropic* [WS01a]. Flow-driven regularisers lead to nonlinear diffusion processes.

Table 2.9 gives an overview of the different regularisers and their corresponding diffusion filters. As a rule of thumb, one can expect that flow-driven regularisers offer advantages over image-driven ones for highly textured sequences, where the numerous texture edges create an oversegmentation of the flow field. Moreover, anisotropic methods may give somewhat better results than isotropic ones, since the latter ones are too "lazy" at noisy discontinuities.

2.3.3 Spatial Regularisation - Experiments

In the previous section, we have classified five strategies for smoothness terms by their induced diffusion process. Let us now evaluate their impact on the quality of the computed flow field with two experiments.

2.3.3.1 Experiment I: Preservation of Discontinuities - Visual Quality

In this experiment, we compare the capabilities of the different smoothness terms regarding the preservation of discontinuities. To this end, we used a grey-scale variant of the synthetic *New Marble* sequence by Middendorf. It consists of 200 frames of size 512×384 pixels and depicts two marbled blocks moving over a marbled floor. A third, stationary block, completes the scene. The sequence as well as the ground truth are available from `http://i21www.ira.uk.de/image_sequences`. For the computation of the flow fields, we considered a spatial approach with data term M_1 (brightness constancy). In contrast to previous experiments, the obtained results were not optimised with with respect to the average angular error, but with respect to visual quality. This allows to investigate the actual behaviour of the different smoothness concepts in a much better way. Apart from the full flow fields depicted in Figure 2.12, also zoom-ins for the corner of the upper right moving block are shown (cf. 2.13). Both illustrations show clearly the following facts:

- *Usefulness of Discontinuity-Preserving Smoothness Terms.* As one can see, the homogeneous regulariser S_1 yields the worst performance of all smoothness terms. Its flow field is fairly blurry and does not respect important motion discontinuities. All other smoothness terms, however, give significantly better results: They allow the preservation of discontinuities for both the small and the large marbled block.
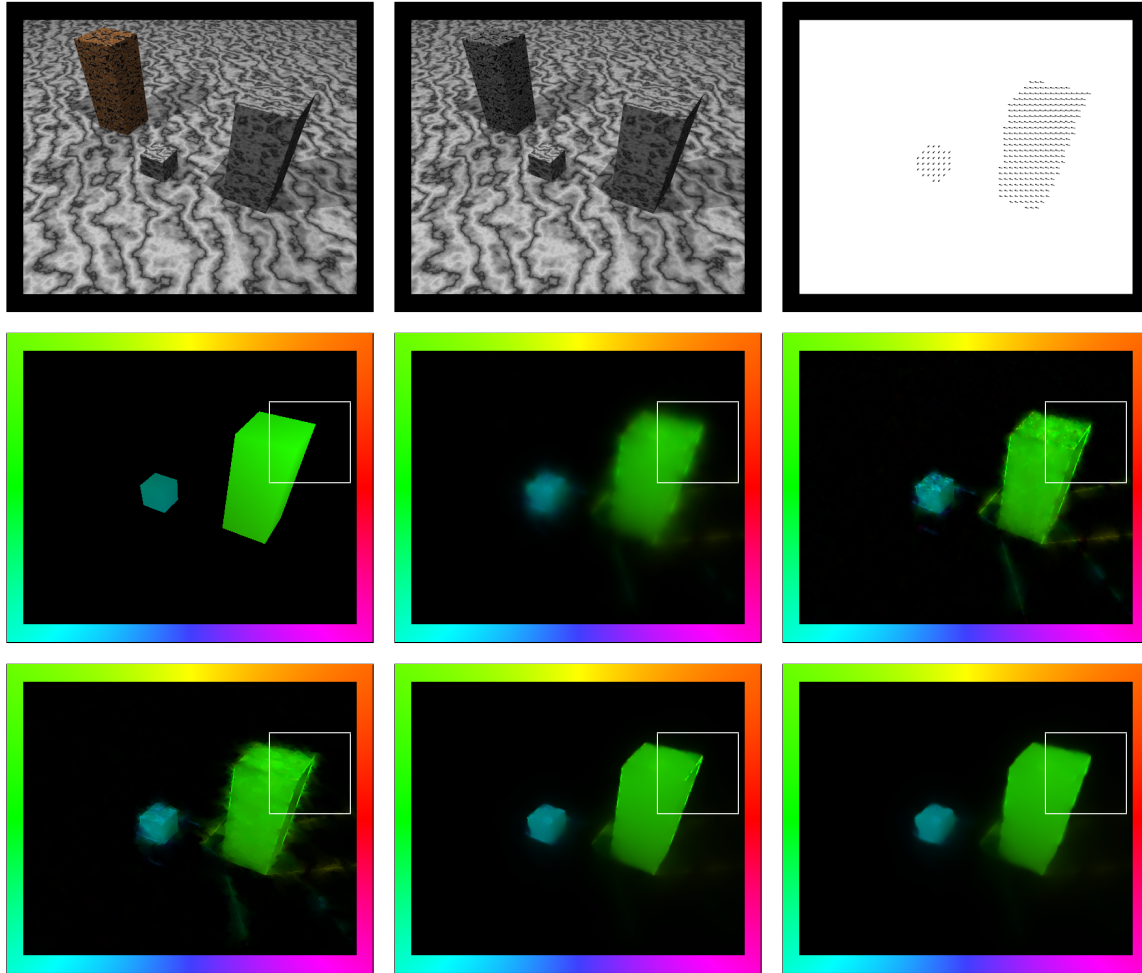
Fig. 2.12:   **Experiment I**: Impact of different smoothness assumptions on the *visual quality* of the flow field with respect to the *preservation of discontinuities* (1/2). **(a) Top Left:** Frame 150 of the *New Marble* sequence of size 512×384 pixels (colour variant). **(b) Top Centre:** The same frame of the converted sequence (grey-scale variant). **(c) Top Right:** Ground truth (vector plot). **(d) Middle Left:** Ground truth (colour plot). **(b) Middle Centre:** Computed optic flow magnitude for a spatial approach with data term M_1 (brightness constancy) and smoothness term S_1 (homogeneous regularisation). **(c) Middle Right:** Smoothness term S_2 (image-driven isotropic regularisation). **(d) Bottom Left:** Smoothness term S_3 (image-driven anisotropic regularisation). **(e) Bottom Centre:** Smoothness term S_4 (flow-driven isotropic regularisation) **(f) Bottom Right:** Smoothness term S_5 (flow-driven anisotropic regularisation).

- *Usefulness of Flow-Driven Smoothness Terms.* Comparing image- and flow-driven regularisers, the superior performance of the flow-driven techniques in highly textured regions becomes obvious. Here, image-driven methods suffer from the fact that they interpret each image discontinuity as motion discontinuity and thus inhibit smoothing. Flow-driven regularisers, however, do not adapt to the image and thus allow for a more homogeneous estimation of the flow field in such regions. This can particularly be seen within the marbled blocks and at their motion boundaries.

- *Usefulness of Anisotropic Smoothness Terms.* Finally, one can also observe that the anisotropic regularisers give slightly better results along flow discontinuities than
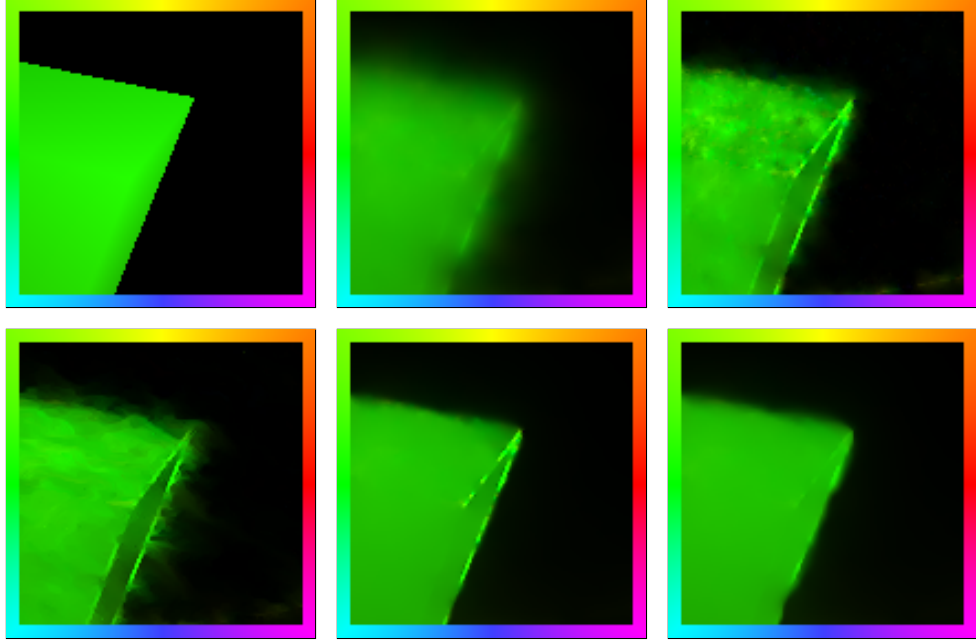
Fig. 2.13:   **Experiment I**: Impact of different smoothness assumptions on the *visual quality* of the flow field with respect to the *preservation of discontinuities* (2/2). **(a) Top Left:** Detail from frame 150 of ground truth of the *New Marble* sequence (128×128 pixels). **(b) Top Centre:** Computed optic flow magnitude for a spatial approach with data term M_1 (brightness constancy) and smoothness term S_1 (homogeneous regularisation). **(c) Top Right:** Smoothness term S_2 (image-driven isotropic regularisation). **(d) Bottom Left:** Smoothness term S_3 (image-driven anisotropic regularisation). **(e) Bottom Centre:** Smoothness term S_4 (flow-driven isotropic regularisation) **(f) Bottom Right:** Smoothness term S_5 (flow-driven anisotropic regularisation).

their isotropic counterparts. This shows the advantage of additionally imposing smoothness along edges, even of smoothness across them is completely inhibited.

This shows that discontinuity-preserving methods should be used, if sharp edges of the flow field are desired. In order to demonstrate the differences between the different regularisers also quantitatively, we performed a second experiment, where we also optimised the parameters with respect to the average angular error.

2.3.3.2   Experiment II: Preservation of Discontinuities - Impact on the Accuracy

Table 2.10 presents our second experiment. It compares the different spatial approaches from our first experiment with respect to their accuracy for the Yosemite sequence *with* clouds. To this end, we computed the average angular error for optimised parameter sets of σ and α. The obtained results confirm the observation we have made in our first experiment:

- *Usefulness of Discontinuity-Preserving Smoothness Terms.* As one can see, the accuracy improves significantly, if adaptive regularisation strategies are used (smoothness terms S_2-S_4). Thereby image and flow-driven techniques perform equally well. This can be explained as follows: If important motion discontinuities coincide with
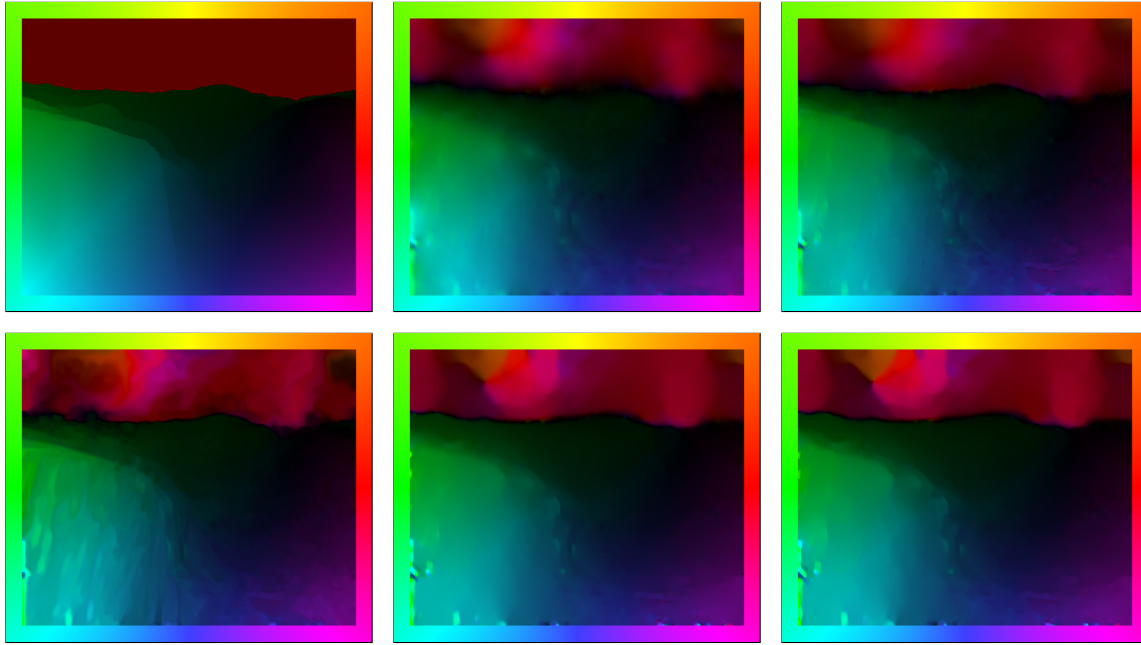
Fig. 2.14:   **Experiment II**: Impact of different smoothness assumptions on the accuracy. of the flow field with respect to the *preservation of discontinuities*. **(a) Top left:** Frame 8 of the *Yosemite* sequence (128×128 pixels). **(b) Top centre:** Computed optic flow magnitude for a spatial approach with data term M_1 (brightness constancy) and smoothness term S_1 (homogeneous regularisation). **(c) Top right:** Smoothness term S_2 (image-driven isotropic regularisation). **(d) Bottom left:** Smoothness term S_3 (image-driven anisotropic regularisation). **(e) Bottom centre:**Smoothness term S_4 (flow-driven isotropic regularisation) **(f) Bottom right:** Smoothness term S_5 (flow-driven anisotropic regularisation).

> image-discontinuities – this is the case at the horizon – image-driven smoothness terms allow for a similar precision as flow-driven regularisers. If, however, as in the first experiment, objects and background are heavily textured, the obtained flow fields suffer from oversegmentation artifacts and the accuracy of flow-driven methods is higher.

The corresponding flow fields depicted in Figure 2.14 are fully in accordance with our considerations: The separation of sky and mountain region is much sharper if discontinuity preserving smoothness terms are used.

2.3.4   Spatiotemporal Regularisation - Modelling

While our general functional (1.3) allows either spatial or spatiotemporal models, the regularisers that we have discussed so far use only *spatial* smoothness constraints. However, in particular with respect to the fact that the motion of objects often varies only slowly over time, it seems desirable to impose some amount of *temporal* or *piecewise temporal* smoothness as well. In this context, one may also think of extending the Gaussian presmoothing to the temporal domain. Let us now investigate what happens if we consider such spatiotemporal models.

   Going from spatial to spatiotemporal models is not very difficult in principle: All one has to do is to replace the spatial integration domain Ω_2 in (1.3) by a spatiotemporal one

Tab. 2.10: **Experiment II**: Impact of different smoothness assumptions on the accuracy. of the flow field with respect to the *preservation of discontinuities*. We used a spatial energy functional with data term M_1 (brightness constancy, and computed the average angular error (AAE) for the Yosemite sequence *with* clouds. The parameters σ and α have been optimised.

smoothness term	regularisation strategy	σ	α	AAE
S_1	homogeneous	1.30	500	7.17°
S_2	image-driven isotropic	1.20	2700	6.44°
S_3	image-driven anisotropic	1.30	4500	6.28°
S_4	flow-driven isotropic	1.30	42	6.32°
S_5	flow-driven anisotropic	1.30	44	6.42°

Ω_3, and to consider spatiotemporal instead of spatial derivatives. As a resulting steepest descent method, one obtains the *three-dimensional* diffusion–reaction system

$$\partial_t u_1 \;=\; \partial_{x_1} S_{u_1,x_1} + \partial_{x_2} S_{u_1,x_2} + \partial_{x_3} S_{u_1,x_3} - \tfrac{1}{\alpha}\ \partial_{u_1} M, \qquad (2.53)$$

$$\partial_t u_2 \;=\; \underbrace{\partial_{x_1} S_{u_2,x_1} + \partial_{x_2} S_{u_2,x_2} + \partial_{x_3} S_{u_2,x_3}}_{\text{diffusion}} - \tfrac{1}{\alpha}\ \underbrace{\partial_{u_2} M}_{\text{reaction}} \qquad (2.54)$$

instead of its two-dimensional counterpart (2.41)–(2.42).

In practice, spatiotemporal models have not been used too often so far. An early suggestion for spatiotemporal anisotropic image-driven regularisers goes back to Nagel [Nag90], followed by spatiotemporal flow-driven approaches such as [BA91, WS01b]. The main reason why such techniques have hardly been studied in the literature is the large amount of memory that is required to process multiple frames simultaneously. In the meantime, however, the fast development of standard desktop PCs allows even the computation of whole image sequences of reasonable size (e.g. of size $30 \times 512 \times 512$ in [BWS05]). Since the computational time per frame is only marginally higher for spatiotemporal methods than for their spatial counterparts and the achieved results are usually significantly better, spatiotemporal methods became increasingly appealing in the last years.

2.3.5 Spatiotemporal Regularisation - Experiments

After we have seen how to model spatiotemporal smoothness terms, let us now analyse their impact on the quality of the computed flow field. To this end, we have performed two experiments.

2.3.5.1 Experiment I: Spatial vs. Spatiotemporal Regularisation under Noise

In our first experiment depicted in Figure 2.15 we study the effect of replacing spatial by spatiotemporal regularisation. This is done by the example of the *Old Marble* sequence by Otte and Nagel (`http://i21www.ira.uk.de/image_sequences`). This real-world sequence consists of 32 frames of size 512×512 pixels and shows four dark marbled blocks that move along with the marbled floor to the lower right. In contrast to all other objects, the white marble block in the centre stands still. Apart from this sequence in its

Tab. 2.11: **Experiment I**: Comparison of the spatial and spatiotemporal regularisers with and without noise. We used a sptial/spatiotemporal energy functional with with data term M_1 (brightness constancy) and smoothness term S_4 (flow-driven isotropic regularisation. We computed the average angular error (AAE) for the *Marble* sequence without and with Gaussian noise with standard deviation $\sigma_n = 40$. The parameters σ_{spat}, σ_{temp} and α have been optimised.

noise	smoothness term	strategy	σ_{spat}	σ_{temp}	α	AAE
$\sigma_n = 0$	S_4 (flow-driven isotropic)	spatial	2.60	-	40	5.11°
	S_4 (flow-driven isotropic)	spatiotemporal	2.00	0.45	13	2.51°
$\sigma_n = 40$	S_4 (flow-driven isotropic)	spatial	2.15	-	400	7.75°
	S_4 (flow-driven isotropic)	spatiotemporal	3.15	0.33	75	4.90°

original form, we also considered a noisy variant with Gaussian noise of standard deviation $\sigma_n = 40$. This allows us to investigate the advantages of spatiotemporal regularisers also with respect to corrupted data. For computing the optic flow, we used an energy functional with data term M_1 (brightness constancy) and spatial or spatiotemporal smoothness term S_4 (flow-driven isotropic regularisation). In order to show significant differences between both types of smoothness terms it was already sufficient to consider four instead of two frames in the spatiotemporal case. Moreover, the input images were also presmoothed also along the temporal direction in order to investigate the impact of spatiotemporal presmoothing on the results. To this end, a Gaussian with different standard deviation in space and time has been used.

The obtained results for the sequence with and without noise are listed in Table 2.11. The raw numbers show the following:

- *Usefulness of Spatiotemporal Regularisers.* Obviously, the results can be significantly improved if a spatiotemporal regulariser is used instead of a spatial one. In the case of the original sequence, one can even observe an improvement of more than 50%. Thereby one should keep in mind that in contrast to the spatial setting only two additional frames have been used. This strong improvement in terms of the average angular error is a consequence of the additional temporal information that allows for a more precise estimation of the optic flow for both stationary and moving objects. While in the case of stationary regions the beneficial effect is obvious – the zero flow is propagated along the temporal axis – things are a bit more complicated for moving areas: In this case, the results can be expected to be smooth in the direction of the flow but not necessarily in the direction of time. If the motion, as in this chapter, is restricted to small displacements and the objects are relatively large, this is not a problem: Both directions almost coincide and any spatiotemporal regulariser will work well. However, if the displacements become sufficiently large with respect to the object size, motion discontinuities in temporal direction will arise. Then, discontinuity preserving smoothness terms such as those proposed with S_2-S_5 have to be considered. One should note that these regularisers do not only allow the preservation of motion boundaries in spatial direction. In their spatiotemporal setting, they also respect discontinuities in direction of the temporal axis.
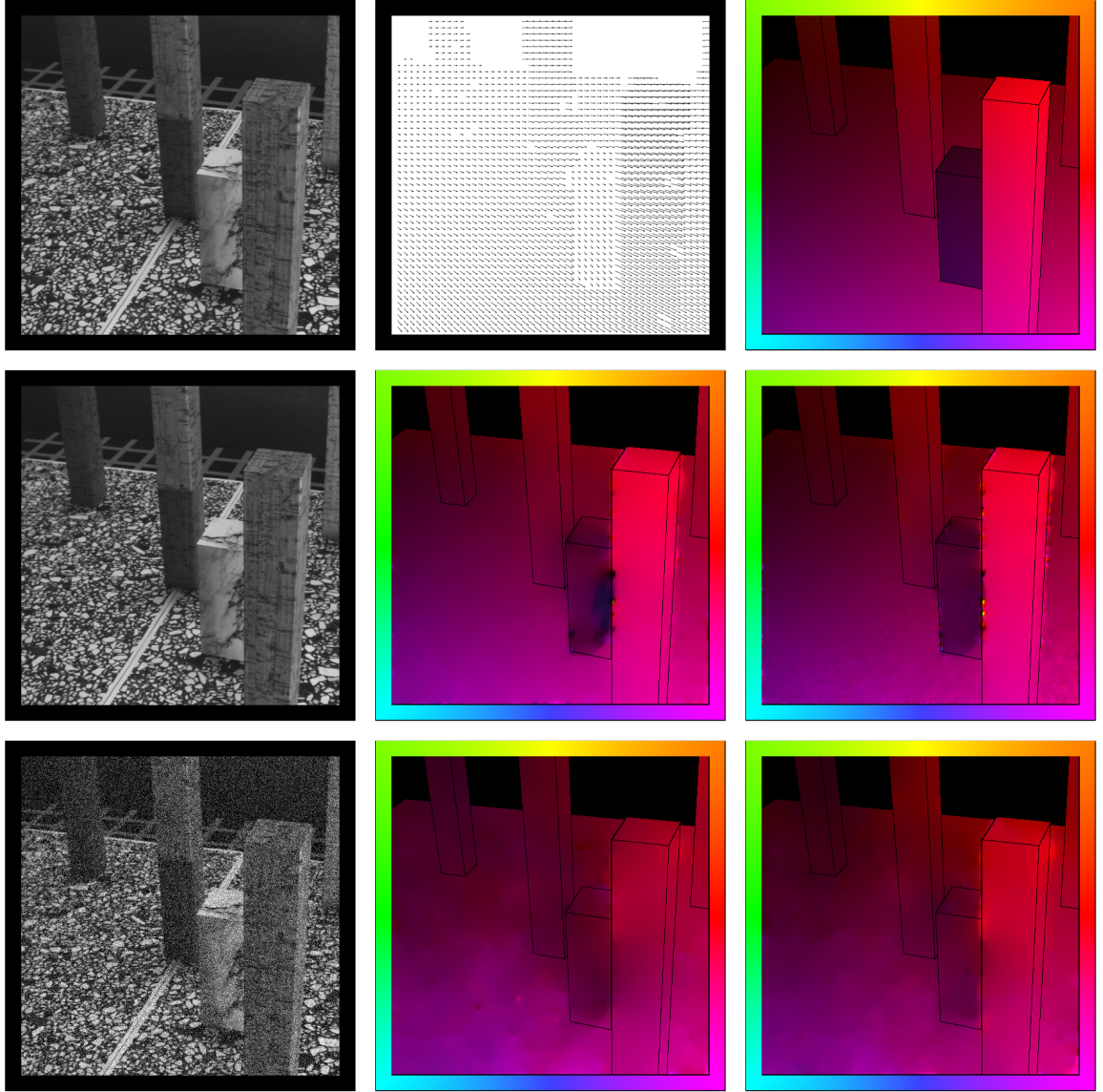
Fig. 2.15: **Experiment I**: Comparison of spatial and spatiotemporal regularisers with and without under noise. **(a) Top Left:** Frame 16 of the *Marble* sequence of size 512×512 pixels. **(b) Top Centre:** Ground truth (vector plot). **(c) Top Right:** Ground truth (colour plot). **(d) Middle Left:** Frame 17 with noise level $\sigma_n = 0$. **(e) Middle Centre:** Computed flow field for a spatial approach with data term M_1 (brightness constancy) and smoothness term S_4 (flow-driven isotropic regularisation). Noise level $\sigma_n = 0$. **(f) Middle Right:** Spatiotemporal approach. **(g) Bottom Left:** Frame 17 with noise level $\sigma_n = 40$ **(h) Bottom Middle:** Spatial approach. **(i) Bottom Right:** Spatiotemporal approach.

- *Usefulness of Spatiotemporal Regularisers with Respect to Noise.* As one can see from the achieved angular errors for the noisy data, the usefulness of spatiotemporal smoothness terms is not only limited to the noise-free case. Also for image data that is corrupted by Gaussian noise, the additional temporal information allows obviously a more reliable estimation of the results.

This experiment shows that significantly better results can be obtained, if spatial regularisers are replaced by their spatiotemporal counterparts. Let us study in our last experiment how the size of the temporal window, i.e. the number of frames, influences the quality of the computed flow field.

Tab. 2.12: **Experiment II**: Impact of the number of frames and the amount of spatiotemporal presmoothing on the quality of the optic flow field. We used a spatiotemporal energy functional with data term M_1 (brightness constancy) and smoothness term S_4 (flow-driven regularisation) and computed the average angular error (AAE) for the Yosemite sequence with clouds. The parameters σ_{spat}, σ_{temp} and α have been optimised.

number of frames	frames	α	spatial presmoothing		spatiotemporal presmoothing	
			σ_{spat}	AAE	σ_{temp}	AAE
2	7-8	500	1.30	7.17°	-	7.17°
3	7-9	340	1.20	6.95°	0.40	6.83°
4	6-9	270	1.30	6.66°	0.45	6.54°
5	6-10	220	1.30	6.70°	0.55	6.56°
6	5-10	255	1.30	6.36°	0.50	6.23°
7	5-11	220	1.30	6.57°	0.50	6.46°
8	4-11	240	1.30	6.44°	0.50	6.31°
9	4-12	190	1.30	6.54°	0.50	6.41°
10	3-12	220	1.30	6.41°	0.50	6.30°
11	3-13	205	1.30	6.48°	0.50	6.37°
12	2-13	205	1.30	6.37°	0.50	6.25°
13	2-14	205	1.30	6.43°	0.50	6.33°
14	1-14	205	1.30	6.37°	0.50	6.25°
15	1-15	180	1.30	6.40°	0.50	6.29°

2.3.5.2   Experiment II: The Spatiotemporal Window

In order to investigate the impact of the size of the spatiotemporal window on the result, we have considered once more the Yosemite sequence *with* clouds. Starting from the centre frames 8 and 9 we have increased the total number of frames that are used for the estimation consequently by one until the maximum number of 15 frames was reached. In order to isolate also the effect of the additional presmoothing along the temporal axis, we computed the results with purely spatial and with spatiotemporal presmoothing. The achieved average angular errors are listed in Table 2.12. They show the following two tendencies:

- *Usefulness of a Small Spatiotemporal Windows*. While increasing the number of input frames of 2 to 6 yields a significant improvement of the results, the use of more than 6 frames does not seem to give additional information that is valuable. However, we can also see that more frames do not necessarily spoil the result, in particular, if discontinuity-preserving spatiotemporal regularisers are used. Moreover, one can observe that the optimal value for the smoothness weight α decreases when the number of frames increases. This is a consequence of the increasing contribution of the temporal information that joins the spatial one by filling in useful

data from the neighbourhood. If there is enough temporal information, one can chose α much smaller but still obtains a similar extent of the *filling-in effect* as in the spatial case. One should note that is particularly useful, since discontinuities are better preserved for smaller values of alpha.

- *Usefulness of Spatiotemporal Presmoothing.* One can furthermore observe that it also makes sense to extend the presmoothing step to the temporal domain. As mentioned in Section 2.1 this step does not only help to eliminate noise: If the sequence is presmoothed with a Gaussian it becomes $f \in \mathcal{C}^\infty$. Moreover, since the convolved sequence changes more smoothly along the spatial and temporal direction, such a presmoothing step allows a better linearisation by means of a Taylor expansion. However, one should note that the computed flow field fits then best to the presmoothed data. Thus it is not surprising, that the optimised values for σ_{temp} are not too large.

2.4   Summary

In this chapter we have discussed the design of variational methods that are suitable for *small* displacements. This was done in two steps. First we have reviewed the modelling of the *data term*: We have seen how constancy assumptions can be modified so that they become more robust under illumination changes, invariant under different types of motion, and better performing with respect to noise and outliers in the data. In the second part we have investigated the design of the *smoothness term*. By identifying different regularisation strategies with their underlying diffusion process, we showed how such diffusion processes can be classified and how they have to be modelled to preserve discontinuities in the image or the flow field. Moreover, we have discussed how to improve the performance even further by extending such smoothness constraints to the temporal domain.

Apart from this overview over a variety of concepts for variational flow methods, we have also introduced the concepts of *motion* and *diffusion tensors*. As we have seen, these two tensors carry the essential information on the underlying optic flow approach and allow to understand the interaction between the data and the smoothness term in a better way. In the following chapters this tensor notation will prove to be even more useful: It will allow a simple formulation of the associated Euler-Lagrange equations and the development of a general strategy to derive efficient hierarchical numerical schemes (see Chapters 3, 4, 5).

3

Optic Flow For Small Displacements – Prototypes

In the previous chapter we focused on the design of variational optic flow methods for small displacements. To this end, we discussed a variety of concepts for the data and the smoothness term and introduced a suitable notation based on motion and diffusion tensors.

In this chapter, we exploit this general toolkit for the systematic construction of variational techniques: We present seven approaches that shall serve as *prototypes* for both our *qualitative* and our *numerical* studies. To this end, we start with a brief discussion of the underlying energy functionals in Section 3.1. Apart from recalling the integrated concepts – they have already been presented in the previous chapter – we also draw connections to similar methods in the literature. This discussion is followed by a comparison of the corresponding Euler-Lagrange equations in Section 3.2. Such a comparison allows us to classify the different equation types into three main groups and relate them to similar types of equation systems. This is done in Section 3.3, where a suitable discretisation based on a finite difference approximation is discussed. Finally, we conclude this chapter with an extensive evaluation of all presented methods: In Section 3.4 various test scenarios shall allow to access the actual *quality* of the different approaches.

3.1 Selection of Prototypes

It should be evident that our framework in Chapter 2 allows the construction of many more variational methods, that we can actually consider in our qualitative and numerical studies. Therefore, we restrict ourselves at this point to seven prototypes that cover most of the proposed concepts. These prototypes can be divided into two different groups: *basic variational approaches* and *advanced variational methods*. For the group of basic approaches we have chosen five spatial methods based on the standard data term M_1 (brightness constancy) and the different regularisers S_1-S_5. This selection of methods based on different regularisation strategies is particularly interesting from a numerical point of view, since each smoothness term requires a slightly different numerical scheme. For the group of advanced variational methods, we have considered two techniques, that we have recently presented: the noise robust CLG method of Bruhn *et al.* [BWS05] and the illumination robust technique of Papenberg *et al.* [PBB$^+$06, BBPW04]. In contrast to the first approaches these methods combine several concepts within one energy functional: While the method of Bruhn *et al.* combines two robust concepts (local least square fit + non-quadratic penalisation), the approach of Papenberg *et al.* accounts for the problem of varying illumination by using a combined data term with two different constancy assumptions. Moreover, both methods are equipped with a discontinuity preserving regulariser. Since we are interested

in having both a spatial and a spatiotemporal technique within this group, we decided to investigate the approach of Papenberg *et al.* in its spatiotemporal form. After this short motivation, let us start our discussion on the prototypes by introducing the representatives of our first group: the basic variational methods.

3.1.1 Basic Variational Methods

As we have seen in the previous chapter, there are basically five different types of smoothness terms: *homogeneous regularisation*, *image-driven isotropic,* and *image-driven anisotropic* regularisation and *flow-driven isotropic* and *flow-driven anisotropic* regularisation [WS01a]. For each of the five strategies we have chosen one prototype based on the motion tensor formulation for the linearised brightness constancy assumption (M_1). In the following these prototypes are discussed briefly.

3.1.1.1 Prototype A – Homogeneous Regularisation

Our prototype for the class of methods with *homogeneous* regularisation is the classical method of Horn and Schunck [HS81]. Their method assumes global smoothness by penalising deviations from smoothness in a quadratic way [TA77]. This comes down to combining the data term M_1 with the smoothness term S_1. The corresponding energy functional reads

$$
\begin{aligned}
E_{\text{HOM}-\text{2D}}(\mathbf{u}) \;&=\; \int_{\Omega_2} \Big(M_1(D^1 f, \mathbf{u}) + \alpha\, S_1(\nabla f, \nabla \mathbf{u}) \Big)\, d\mathbf{x} \\
&=\; \int_{\Omega_2} \Big((\mathbf{u}^\top J_1\, \mathbf{u} + \alpha\, \big(\, |\nabla u_1|^2 + |\nabla u_2|^2\, \big)\, \Big)\, d\mathbf{x},
\end{aligned}
\tag{3.1}
$$

where $J_1 = \nabla_3 f \nabla_3 f^\top$ is the motion tensor associated to the brightness constancy assumption as defined in the last chapter.

3.1.1.2 Prototype B – Image-Driven Isotropic Regularisation

Instead of penalising deviations from smoothness in a quadratic way, one may think of downweighting the smoothness term at locations where the magnitude of the spatial image gradient is large [AELS99]. As we have discussed in Chapter 2, this form of regularisation that respects discontinuities in the image data is called *image-driven isotropic*. The associated energy functional is given by

$$
\begin{aligned}
E_{\text{II}-\text{2D}}(\mathbf{u}) \;&=\; \int_{\Omega_2} \Big(M_1(D^1 f, \mathbf{u}) + \alpha\, S_2(\nabla f, \nabla \mathbf{u}) \Big)\, d\mathbf{x} \\
&=\; \int_{\Omega_2} \Big(\mathbf{u}^\top J_1\, \mathbf{u} + \alpha\, g(|\nabla f|^2)\, \big(\, |\nabla u_1|^2 + |\nabla u_2|^2\, \big) \Big)\, d\mathbf{x},
\end{aligned}
\tag{3.2}
$$

where $g(s^2)$ is a positive decreasing function in $\mathbb{R}$. The method we have chosen to represent this regularisation strategy is based on a function proposed by Charbonnier [CBFAB94] which

reads

$$g(s^2) = \frac{1}{\sqrt{1 + \frac{s^2}{\epsilon_S^2}}}. \tag{3.3}$$

Here, ϵ_S is a contrast parameter.

3.1.1.3 Prototype C – Image-Driven Anisotropic Regularisation

As prototype for the class of optic flow methods with *image-driven anisotropic* regularisation we consider the technique of Nagel and Enkelmann [NE86]. Their method accounts for the problem of discontinuities by smoothing only along a projection of the flow gradient, namely its component orthogonal to the local image gradient. As a consequence, flow fields are obtained that avoid smoothing across discontinuities in the image data. The energy functional associated to this anisotropic form of regularisation is given by

$$
\begin{aligned}
E_{\mathrm{IA-2D}}(\mathbf{u}) &= \int_{\Omega_2} \Big(M_1(D^1 f, \mathbf{u}) + \alpha\, S_3(\nabla f, \nabla \mathbf{u}) \Big)\, d\mathbf{x} \tag{3.4} \\
&= \int_{\Omega_2} \Big(\mathbf{u}^\top J_1\, \mathbf{u} + \alpha \big(\nabla u_1^\top P_{\mathrm{NE}}(\nabla f) \nabla u_1 + \nabla u_2^\top P_{\mathrm{NE}}(\nabla f) \nabla u_2 \big) \Big)\, d\mathbf{x},
\end{aligned}
$$

where $P_{\mathrm{NE}}(\nabla f)$ is a projection matrix perpendicular to ∇f that is defined as

$$P_{\mathrm{NE}}(\nabla f) = \frac{1}{|\nabla f|^2 + 2\epsilon_S^2} \begin{pmatrix} f_{x_2}^2 + \epsilon_S^2 & -f_{x_1} f_{x_2} \\ -f_{x_1} f_{x_2} & f_{x_1}^2 + \epsilon_S^2 \end{pmatrix}. \tag{3.5}$$

In this context ϵ_S serves as regularisation parameter that prevents the matrix $P_{\mathrm{NE}}(\nabla f)$ from getting singular.

3.1.1.4 Prototype D – Flow-Driven Isotropic Regularisation

Our fourth method serves as prototype for *flow-driven isotropic* techniques [Sch94b]. By reducing smoothing at those locations where edges in the flow field occur during the computation, it avoids the typical oversegmentation artifacts of image-driven methods. The corresponding energy functional reads

$$
\begin{aligned}
E_{\mathrm{FI-2D}}(\mathbf{u}) &= \int_{\Omega_2} \Big(M_1(D^1 f, \mathbf{u}) + \alpha\, S_4(\nabla f, \nabla \mathbf{u}) \Big)\, d\mathbf{x} \tag{3.6} \\
&= \int_{\Omega_2} \Big(\mathbf{u}^\top J_1\, \mathbf{u} + \alpha\, \Psi_S \big(|\nabla u_1|^2 + |\nabla u_2|^2 \big) \Big)\, d\mathbf{x},
\end{aligned}
$$

where $\Psi_s(s^2)$ is a positive increasing function in $\mathbb{R}$ with the property to increase less severely than a quadratic function. As prototype we have chosen a method that penalises deviations from the smoothness with the L_1 norm. This corresponds to total variation (TV) regularisation [ROF92] which we implemented by means of a regularised variant given by

$$\Psi_S(s^2) = \sqrt{s^2 + \epsilon_S^2}. \tag{3.7}$$

Here ϵ_S serves as small regularisation parameter. A similar functional that approximates TV regularisation is proposed in [WS01b], while variational approaches for rotationally not invariant versions of TV regularisation have been investigated in [Coh93, DKA95, KTB96, ADK99].

3.1.1.5   Prototype E – Flow-Driven Anisotropic Regularisation

The fifth and last regularisation strategy are *flow-driven anisotropic* smoothness terms [WS01a]. In contrast to the isotropic case where the non-quadratic function Ψ_s penalises the magnitude of the flow vector, it is now applied to the local flow tensor $\nabla u_1 \nabla u_1^\top + \nabla u_2 \nabla u_2^\top$ which additionally contains directional information. As we have seen in the previous chapter, the application of the function to a tensor can be realised by means of an eigenvalue decomposition. This allows to access the eigenvalues directly and thus yields an penalisation that adapts to the local flow structure. The associated energy functional is given by

$$
\begin{aligned}
E_{\mathrm{FA-2D}}(\mathbf{u}) &= \int_{\Omega_2} \Big(M_1(D^1 f, \mathbf{u}) + \alpha\, S_5(\nabla f, \nabla \mathbf{u}) \Big)\, d\mathbf{x} & (3.8) \\
&= \int_{\Omega_2} \Big(\mathbf{u}^\top J_1\, \mathbf{u} + \alpha\, \mathrm{tr}\, \big(\Psi_\mathrm{S}(\, \nabla u_1 \nabla u_1^\top + \nabla u_2 \nabla u_2^\top\,)\big) \Big)\, d\mathbf{x},
\end{aligned}
$$

where tr is the trace of the local flow tensor. As for the isotropic case we have chosen a method as prototype that is based on a regularised variant of the total variation penaliser:

$$
\Psi_\mathrm{S}(s^2) = \sqrt{s^2 + \epsilon_\mathrm{S}^2}. \tag{3.9}
$$

3.1.2   Advanced Variational Approaches

After having introduced our prototypes for the five types of regularisation strategies, let us now discuss two prototypes for more advanced optic flow techniques: The noise robust combined-local-global (CLG) approach of Bruhn *et al.* that we presented in [BWS05] and a *linearised variant* of the highly accurate optic flow method of Papenberg *et al.* that we recently proposed in [PBB$^+$06]. In contrast to all previous prototypes these methods combine several of the concepts proposed in the second chapter and can thus be expected to give qualitatively better results. In the following both techniques are explained in detail.

3.1.2.1   Prototype F – The Noise Robust Method of Bruhn *et al.*

(f) Noise Robustness - The Method of Bruhn *et al.*

As we have seen in Section 2.2.5, the sensitivity of approaches with respect to noise is a very important aspect for the design of algorithms in motion estimation. Therefore, we present a variational approach that tackles this problem in two ways: (i) It combines the *noise robustness* of local methods with the full density of global approaches. This is achieved by embedding a local least square fit into the motion tensor formulation of the data term [BWS02]. How this works in detail has been shown in Section 2.2.5.1. (ii) It makes use of the regularised L_1 norm as non-quadratic penaliser of the data term. As we have discussed in Section 2.2.5.3, such non-quadratic functions are related to statistically

robust error norms [Hub81] and thus increase the performance of the approach with respect to *outliers* [BA91]. Apart from these two concepts that shall improve the performance of the data term, we also consider a discontinuity preserving smoothness term: a flow-driven isotropic regulariser based on the regularised total variation (TV). The associated energy functional for this noise robust method is given by

$$
\begin{aligned}
E_{\mathrm{CLG-2D-SD}}(\mathbf{u}) &= \int_{\Omega_2} \Big(M_{12}(D^1 f, \mathbf{u}) + \alpha\, S_4(\nabla f, \nabla \mathbf{u})\, d\mathbf{x} \qquad\qquad (3.10) \\
&= \int_{\Omega_2} \Big(\Psi_{\mathrm{D}}\left(\mathbf{u}^\top J_9\, \mathbf{u}\right) + \alpha\, \Psi_{\mathrm{S}}\big(\, |\nabla u_1|^2 + |\nabla u_2|^2 \,\big) \Big)\, d\mathbf{x}.
\end{aligned}
$$

As for our previous prototypes the small constants required for the regularised L_1 norm are denoted by ϵ_{D} and ϵ_{S} for the data and the smoothness term, respectively. Please recall that the motion J_9 is obtained via Gaussian convolution from J_1: $J_9(\nabla_3 f) = K_\rho * J_1(\nabla_3 f)$. In this context, one should also note that the subscript SD stands for "small displacements". This refers to the fact, that only linearised constancy assumptions are used.

3.1.2.2 Prototype G – The Illumination Robust Method of Papenberg *et al.*

Apart from noise there are is another problem that has to be addressed in the context of motion estimation: varying illumination. To this end, we propose a variational approach that supplements the standard grey value constancy assumption by an additional term: The constancy of the spatial image gradient $\nabla_2 f = (f_{x_1}, f_{x_2})^\top$. As we have seen in the previous chapter, this assumption allows to deal with global illumination changes of additive type. However, we have also seen that this assumption is very sensitive with respect to noise. Therefore, we render the data term more robust with respect to outliers by a joint robustification of both constancy assumptions. To this end, a regularised variant of the L_1 norm is used. Finally, since we are also interested in the preservation of motion discontinuities, we consider a spatiotemporal flow-driven isotropic regulariser based on the regularised total variation (TV). Combining all these concepts within a single approach yields the following energy functional:

$$
\begin{aligned}
E_{\mathrm{PAP-3D-SD}}(\mathbf{u}) &= \int_{\Omega_3} \Big(M_{13}(D^1 f, D^2 f, \mathbf{u}) + \alpha\, S_4(\nabla f, \nabla \mathbf{u})\, d\mathbf{x} \qquad\quad (3.11) \\
&= \int_{\Omega_3} \Big(\Psi_{\mathrm{D}}\left(\mathbf{u}^\top J_7\, \mathbf{u}\right) + \alpha\, \Psi_{\mathrm{S}}\big(\, |\nabla_3 u_1|^2 + |\nabla_3 u_2|^2 \,\big) \Big)\, d\mathbf{x}.
\end{aligned}
$$

As defined in the previous chapter the motion tensor J_7 is given by

$$
\begin{aligned}
J_7(\nabla_3 f) &= \gamma_1 J_1(\nabla_3 f) + \gamma_2 J_2(\nabla_3 f_{x_1}, \nabla_3 f_{x_2}) \\
&= \gamma_1 \Big(\nabla_3 f \nabla_3 f^\top \Big) + \gamma_2 \Big(\nabla_3 f_{x_1} \nabla_3 f_{x_1}^\top + \nabla_3 f_{x_2} \nabla_3 f_{x_2}^\top \Big).
\end{aligned}
$$

where J_1 and J_2 are the motions tensor for the brightness and the gradient constancy assumption, respectively. Since in general, only the relative weight between both terms is decisive for the estimation, one may set one γ_i fixed. In our case we decided to set $\gamma_1 = 1$ and to replace γ_2 by γ.

Please note that this prototype is a *variant* of the original method of Papenberg that we presented in *et al.* [PBB$^+$06]: Due to the use of linearised constancy assumptions this variant is limited to the estimation of small displacements.

3.2   Minimisation

After we have presented our prototypes, let us now discuss how the underlying energy functionals can be minimised. As we have seen in the previous chapter, this can be done in two ways: Either one computes the solution of the corresponding Euler–Lagrange equations (*elliptic approach*) or one finds the steady-state of the associated diffusion-reaction system that is given by the steepest descent equations (*parabolic approach*). In the following we restrict ourselves to the elliptic case. As experiments in our numerical section will show, this strategy is more efficient than the parabolic one.

3.2.1   The Euler–Lagrange Equations

According to the Calculus of Variations [Els61], the minimisation of an energy functionals comes down to solving their Euler–Lagrange equations. In order to solve them, however, these equations have first to be formulated. By using the motion and diffusion tensor notation that we have introduced in the last chapter, this can be done in a compact way. Moreover, this systematic notation can be exploited to derive a classification of the different prototypes. This classification is based on the type of the associated Euler–Lagrange equations. Thereby three cases are distinguished: the *linear case*, the *nonlinear case I* (still partially linear) and the *nonlinear case II* (fully nonlinear). In order to keep the classification general, the specific motion tensor for each prototype is replaced in the following by an arbitrary motion tensor J.

3.2.1.1   The Linear Case – Prototypes A-C

In the first three cases A, B and C the Euler–Lagrange equations have the coupled form

$$0 \;=\; J_{11}\,u_1 + J_{12}\,u_2 + J_{13} \;-\; \alpha\,\mathcal{L}_{\mathrm{L}}u_1, \qquad (3.12)$$

$$0 \;=\; \underbrace{J_{12}\,u_1 + J_{22}\,u_2 + J_{23}}_{\text{linear}} \;-\; \alpha\,\underbrace{\mathcal{L}_{\mathrm{L}}u_2}_{\text{linear}} \qquad (3.13)$$

with the *linear* differential operator

$$\mathcal{L}_{\mathrm{L}}z(\mathbf{x}) = \mathrm{div}\,\left(T(\nabla f)\,\nabla z(\mathbf{x})\right) \qquad (3.14)$$

and homogeneous Neumann (reflecting) boundary conditions. This linearity results from the adaptation of the corresponding smoothness terms to the discontinuities in the *image sequence*. The corresponding diffusion tensors are given by

A	- homogeneous regularisation	$T(\nabla f) = I$
B	- image-driven isotropic regularisation	$T(\nabla f) = g(|\nabla f|^2)\,I$
C	- image-driven anisotropic regularisation	$T(\nabla f) = P_{\mathrm{NE}}(\nabla f)$

3.2.1.2   The Nonlinear Case I – Prototypes D-E

In the cases D and E the associated Euler-Lagrange equations have a structure that is very similar to the one for A, B and C. They are given by the coupled form

$$0 = J_{11}\,u_1 + J_{12}\,u_2 + J_{13} - \alpha\,\mathcal{L}_{\mathrm{NL}}(u_1, u_2), \tag{3.15}$$

$$0 = \underbrace{J_{12}\,u_1 + J_{22}\,u_2 + J_{23}}_{\text{linear}} - \alpha\,\underbrace{\mathcal{L}_{\mathrm{NL}}(u_2, u_1)}_{\text{nonlinear}} \tag{3.16}$$

with the *nonlinear* differential operator

$$\mathcal{L}_{\mathrm{NL}}(z(\mathbf{x}), \tilde{z}(\mathbf{x})) = \mathrm{div}\,\left(T(\nabla z(\mathbf{x}), \nabla \tilde{z}(\mathbf{x}))\,\nabla z(\mathbf{x}) \right) \tag{3.17}$$

and homogeneous Neumann boundary conditions. Here, $\mathcal{L}_{\mathrm{NL}}$ is a nonlinear differential operator, because it depends nonlinearly on its arguments z and $\tilde{z}$ (which are in fact u_1 and u_2). This can directly be seen from the corresponding diffusion tensors that are given by

D - flow-driven isotropic regularisation	$T(\nabla z, \nabla \tilde{z}) = \Psi_{\mathrm{S}}'(|\nabla z|^2 + |\nabla \tilde{z}|^2)\,I$
E - flow-driven anisotropic regularisation	$T(\nabla z, \nabla \tilde{z}) = \Psi_{\mathrm{S}}'(\nabla z \nabla z^\top + \nabla \tilde{z} \nabla \tilde{z}^\top)$

where the derivative of the regularised L_1 norm is obviously nonlinear since it reads

$$\Psi_{\mathrm{S}}'(s^2) = \frac{1}{2\sqrt{s^2 + \epsilon_{\mathrm{S}}^2}}\,. \tag{3.18}$$

As we will see later, this nonlinearity of the differential operator $\mathcal{L}_{\mathrm{NL}}$ has serious impact on the resulting discrete system of equations and therewith on the design of the numerical strategy.

Let us now derive the Euler–Lagrange equations for the prototypes of the more advanced optic flow methods F-G.

3.2.1.3   The Nonlinear Case II - Prototypes F-G

In the case of the advanced variational methods for small displacements (Bruhn *et al.*, Papenberg *et al.*), the Euler–Lagrange equations are very similar to those of the flow-driven isotropic method (prototype D). They have basically the same *nonlinear* differential operator and diffusion tensor and are given by the coupled form

$$0 = \Psi_{\mathrm{D}}'\left(\mathbf{u}^\top J\,\mathbf{u}\right)(J_{11}\,u_1 + J_{12}\,u_2 + J_{13}) - \alpha\,\mathcal{L}_{\mathrm{NL}}(u_1, u_2), \tag{3.19}$$

$$0 = \underbrace{\Psi_{\mathrm{D}}'\left(\mathbf{u}^\top J\,\mathbf{u}\right)(J_{12}\,u_1 + J_{22}\,u_2 + J_{23})}_{\text{nonlinear}} - \alpha\,\underbrace{\mathcal{L}_{\mathrm{NL}}(u_2, u_1)}_{\text{nonlinear}}\,. \tag{3.20}$$

Please note that for a spatiotemporal method such as the 3-D method of Papenberg *et al.* (prototype G), the nonlinear differential operator is also *spatiotemporal*, i.e.

$$\mathcal{L}_{\mathrm{NL}}(z(\mathbf{x}), \tilde{z}(\mathbf{x})) = \mathrm{div}\left(T(\nabla_3 z(\mathbf{x}), \nabla_3 \tilde{z}(\mathbf{x}))\, \nabla_3 z(\mathbf{x})\right). \tag{3.21}$$

The corresponding diffusion tensors are given by

$$
\begin{array}{lll}
\text{F} & - & \text{Bruhn } \textit{et al.} \quad T(\nabla z, \nabla \tilde{z}) = \Psi_{\mathrm{S}}'(|\nabla z|^2 + |\nabla \tilde{z}|^2)\, I \\
\text{G} & - & \text{Papenberg } \textit{et al.} \; T(\nabla_3 z, \nabla_3 \tilde{z}) = \Psi_{\mathrm{S}}'(|\nabla_3 z|^2 + |\nabla_3 \tilde{z}|^2)\, I
\end{array}
$$

where the derivative of the regularised L_1 norm in both data and smoothness term reads

$$\Psi_{\mathrm{D}}'(s^2) = \Psi_{\mathrm{S}}'(s^2) = \frac{1}{2\sqrt{s^2 + \epsilon_{\mathrm{S}}^2}}. \tag{3.22}$$

The main difference to the nonlinear case I with prototypes D-E is the fact that the obtained system of equations is now nonlinear in *each* of the terms. The reason for this is the non-quadratic penalisation of outliers in the data term via Ψ_{D} that results in the *nonlinear* factor $\Psi_{\mathrm{D}}'\left(\mathbf{u}^\top J\, \mathbf{u}\right)$ in front of the originally linear expressions J_{nm} $(n, m = 1, .., 3)$.

3.3 Discretisation

3.3.1 Spatial Discretisation Aspects

Let us now discuss a suitable discretisation for the Euler-Lagrange equations A-F. To this end, we consider the unknown functions $u_1(x_1, x_2, x_3)$ and $u_2(x_1, x_2, x_3)$ on a rectangular pixel grid with cell size $\mathbf{h} = (h_{x_1}, h_{x_2})^\top$, and we denote by $\left[u_1^{\mathbf{h}}\right]_{i,j}$ the approximation to u_1 at some pixel i, j with $i = 1, ..., N_{x_1}^{\mathbf{h}}$ and $j = 1, ..., N_{x_2}^{\mathbf{h}}$. The total number of pixels is thus given by $N^{\mathbf{h}} = N_{x_1}^{\mathbf{h}} N_{x_2}^{\mathbf{h}}$. Spatial derivatives of the image data are approximated using a fourth-order approximation with the stencil $(1, -8, 0, 8, -1)/(12h)$, while temporal derivatives are computed with a simple two-point stencil. If we denote the discretised entries of the different *spatial* diffusions tensors by

$$T(\mathbf{x}) := \begin{pmatrix} a(\mathbf{x}) & b(\mathbf{x}) \\ b(\mathbf{x}) & c(\mathbf{x}) \end{pmatrix} \;\rightarrow\; \left[T^{\mathbf{h}}\right]_{i,j} := \begin{pmatrix} \left[a^{\mathbf{h}}\right]_{i,j} & \left[b^{\mathbf{h}}\right]_{i,j} \\ \left[b^{\mathbf{h}}\right]_{i,j} & \left[c^{\mathbf{h}}\right]_{i,j} \end{pmatrix} \tag{3.23}$$

we can discretise the divergence expressions in the spatial differential operators L_{L} and L_{NL} by means of the following finite difference approximations:

$$\partial_{x_1}\left(a(\mathbf{x})\,\partial_{x_1} z(\mathbf{x})\right) \;\approx\; D_{x_1}^{-,\mathbf{h}}\left(M_{x_1}^{+,\mathbf{h}}\left(\left[a^{\mathbf{h}}\right]_{i,j}\right) D_{x_1}^{+,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right)\right), \tag{3.24}$$

$$\partial_{x_1}\left(b(\mathbf{x})\,\partial_{x_2} z(\mathbf{x})\right) \;\approx\; D_{x_1}^{\mathbf{h}}\left(\left[b^{\mathbf{h}}\right]_{i,j} D_{x_2}^{\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right)\right), \tag{3.25}$$

$$\partial_{x_2}\left(b(\mathbf{x})\,\partial_{x_1} z(\mathbf{x})\right) \;\approx\; D_{x_2}^{\mathbf{h}}\left(\left[b^{\mathbf{h}}\right]_{i,j} D_{x_1}^{\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right)\right), \tag{3.26}$$

$$\partial_{x_2}\left(c(\mathbf{x})\,\partial_{x_2} z(\mathbf{x})\right) \;\approx\; D_{x_2}^{-,\mathbf{h}}\left(M_{x_2}^{+,\mathbf{h}}\left(\left[c^{\mathbf{h}}\right]_{i,j}\right) D_{x_2}^{+,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right)\right). \tag{3.27}$$

Details on the required averaging and differential operators within the approximations are given in Table 3.1. One should note, that these discretisations can also be derived from discrete versions of the original energy functionals. An alternative discretisation that also offers a good accuracy can be found in [Bro05].

Tab. 3.1: Discretisations of averaging and differential operators - spatial case.

One-sided averaging	$M_{x_1}^{\pm,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right) := \dfrac{\left[z^{\mathbf{h}}\right]_{i\pm1,j}+\left[z^{\mathbf{h}}\right]_{i,j}}{2}$
	$M_{x_2}^{\pm,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right) := \dfrac{\left[z^{\mathbf{h}}\right]_{i,j\pm1}+\left[z^{\mathbf{h}}\right]_{i,j}}{2}$
One-sided differences	$D_{x_1}^{\pm,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right) := \pm\dfrac{\left[z^{\mathbf{h}}\right]_{i\pm1,j}-\left[z^{\mathbf{h}}\right]_{i,j}}{h_{x_1}}$
	$D_{x_2}^{\pm,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right) := \pm\dfrac{\left[z^{\mathbf{h}}\right]_{i,j\pm1}-\left[z^{\mathbf{h}}\right]_{i,j}}{h_{x_2}}$
Central differences	$D_{x_1}^{\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right) := \dfrac{\left[z^{\mathbf{h}}\right]_{i+1,j}-\left[z^{\mathbf{h}}\right]_{i-1,j}}{2h_{x_1}}$
	$D_{x_2}^{\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right) := \dfrac{\left[z^{\mathbf{h}}\right]_{i,j+1}-\left[z^{\mathbf{h}}\right]_{i,j-1}}{2h_{x_2}}$
Squared differences	$D_{x_1}^{2,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right) := \frac{1}{2}\left(D_{x_1}^{+,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right)\right)^2 + \frac{1}{2}\left(D_{x_1}^{-,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right)\right)^2$
	$D_{x_2}^{2,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right) := \frac{1}{2}\left(D_{x_2}^{+,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right)\right)^2 + \frac{1}{2}\left(D_{x_2}^{-,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right)\right)^2$
Gradient magnitude	$\left|D^{2,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right)\right| := \sqrt{D_{x_1}^{2,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right) + D_{x_2}^{2,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j}\right)}$

3.3.2 Spatiotemporal Discretisation Aspects

In the case of our prototype G (the illumination invariant method of Papenberg *et al.*) we consider the unknown functions $u_1(x_1, x_2, x_3)$ and $u_2(x_1, x_2, x_3)$ on a voxel grid with voxel size $\mathbf{h} = (h_{x_1}, h_{x_2}, h_{x_3})^\top$. Analogously to the spatial case, $\left[u_1^{\mathbf{h}}\right]_{i,j,k}$ stands hereby for the approximation to u_1 at some pixel i, j, k with $i = 1,...,N_{x_1}^{\mathbf{h}}$, $j = 1,...,N_{x_2}^{\mathbf{h}}$, and $k = 1,...,N_{x_3}^{\mathbf{h}}$. This time, however, we have to discretise the differential operator L_{NL} for a *spatiotemporal* diffusion tensor

$$T_3(\mathbf{x}) := \begin{pmatrix} a(\mathbf{x}) & b(\mathbf{x}) & d(\mathbf{x}) \\ b(\mathbf{x}) & c(\mathbf{x}) & e(\mathbf{x}) \\ d(\mathbf{x}) & e(\mathbf{x}) & f(\mathbf{x}) \end{pmatrix} \rightarrow \left[T_{3-D}^{\mathbf{h}}\right]_{i,j,k} := \begin{pmatrix} \left[a^{\mathbf{h}}\right]_{i,j,k} & \left[b^{\mathbf{h}}\right]_{i,j,k} & \left[d^{\mathbf{h}}\right]_{i,j,k} \\ \left[b^{\mathbf{h}}\right]_{i,j,k} & \left[c^{\mathbf{h}}\right]_{i,j,k} & \left[e^{\mathbf{h}}\right]_{i,j,k} \\ \left[d^{\mathbf{h}}\right]_{i,j,k} & \left[e^{\mathbf{h}}\right]_{i,j,k} & \left[f^{\mathbf{h}}\right]_{i,j,k} \end{pmatrix} \quad (3.28)$$

To this end, we use the following finite difference approximations:

$$\partial_{x_1}\left(a(\mathbf{x})\,\partial_{x_1}z(\mathbf{x})\right) \approx D_{x_1}^{-,\mathbf{h}}\left(M_{x_1}^{+,\mathbf{h}}\left(\left[a^{\mathbf{h}}\right]_{i,j,k}\right)D_{x_1}^{+,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right)\right), \qquad (3.29)$$

$$\partial_{x_1}\left(b(\mathbf{x})\,\partial_{x_2}z(\mathbf{x})\right) \approx D_{x_1}^{\mathbf{h}}\left(\left[b^{\mathbf{h}}\right]_{i,j,k}D_{x_2}^{\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right)\right), \qquad (3.30)$$

$$\partial_{x_1}\left(d(\mathbf{x})\,\partial_{x_3}z(\mathbf{x})\right) \approx D_{x_1}^{\mathbf{h}}\left(\left[d^{\mathbf{h}}\right]_{i,j,k}D_{x_3}^{\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right)\right), \qquad (3.31)$$

$$\partial_{x_2}\left(b(\mathbf{x})\,\partial_{x_1}z(\mathbf{x})\right) \approx D_{x_2}^{\mathbf{h}}\left(\left[b^{\mathbf{h}}\right]_{i,j,k}D_{x_1}^{\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right)\right), \qquad (3.32)$$

$$\partial_{x_2}\left(c(\mathbf{x})\,\partial_{x_2}z(\mathbf{x})\right) \approx D_{x_2}^{-,\mathbf{h}}\left(M_{x_2}^{+,\mathbf{h}}\left(\left[c^{\mathbf{h}}\right]_{i,j,k}\right)D_{x_2}^{+,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right)\right), \qquad (3.33)$$

Tab. 3.2: Discretisations of averaging and differential operators - spatiotemporal case.

One-sided averaging	$M_{x_1}^{\pm,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right) := \dfrac{\left[z^{\mathbf{h}}\right]_{i\pm1,j,k}+\left[z^{\mathbf{h}}\right]_{i,j,k}}{2}$
	$M_{x_2}^{\pm,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right) := \dfrac{\left[z^{\mathbf{h}}\right]_{i,j\pm1,k}+\left[z^{\mathbf{h}}\right]_{i,j,k}}{2}$
	$M_{x_3}^{\pm,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right) := \dfrac{\left[z^{\mathbf{h}}\right]_{i,j,k\pm1}+\left[z^{\mathbf{h}}\right]_{i,j,k}}{2}$
One-sided differences	$D_{x_1}^{\pm,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right) := \pm\dfrac{\left[z^{\mathbf{h}}\right]_{i\pm1,j,k}-\left[z^{\mathbf{h}}\right]_{i,j,k}}{h_{x_1}}$
	$D_{x_2}^{\pm,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right) := \pm\dfrac{\left[z^{\mathbf{h}}\right]_{i,j\pm1,k}-\left[z^{\mathbf{h}}\right]_{i,j,k}}{h_{x_2}}$
	$D_{x_3}^{\pm,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right) := \pm\dfrac{\left[z^{\mathbf{h}}\right]_{i,j,k\pm1}-\left[z^{\mathbf{h}}\right]_{i,j,k}}{h_{x_3}}$
Central differences	$D_{x_1}^{\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right) := \dfrac{\left[z^{\mathbf{h}}\right]_{i+1,j,k}-\left[z^{\mathbf{h}}\right]_{i-1,j,k}}{2h_{x_1}}$
	$D_{x_2}^{\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right) := \dfrac{\left[z^{\mathbf{h}}\right]_{i,j+1,k}-\left[z^{\mathbf{h}}\right]_{i,j-1,k}}{2h_{x_2}}$
	$D_{x_3}^{\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right) := \dfrac{\left[z^{\mathbf{h}}\right]_{i,j,k+1}-\left[z^{\mathbf{h}}\right]_{i,j,k-1}}{2h_{x_2}}$
Squared differences	$D_{x_1}^{2,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right) := \frac{1}{2}\left(D_{x_1}^{+,\mathbf{h}}(\left[z^{\mathbf{h}}\right]_{i,j,k})\right)^2 + \frac{1}{2}\left(D_{x_1}^{-,\mathbf{h}}(\left[z^{\mathbf{h}}\right]_{i,j,k})\right)^2$
	$D_{x_2}^{2,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right) := \frac{1}{2}\left(D_{x_2}^{+,\mathbf{h}}(\left[z^{\mathbf{h}}\right]_{i,j,k})\right)^2 + \frac{1}{2}\left(D_{x_2}^{-,\mathbf{h}}(\left[z^{\mathbf{h}}\right]_{i,j,k})\right)^2$
	$D_{x_3}^{2,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right) := \frac{1}{2}\left(D_{x_3}^{+,\mathbf{h}}(\left[z^{\mathbf{h}}\right]_{i,j,k})\right)^2 + \frac{1}{2}\left(D_{x_3}^{-,\mathbf{h}}(\left[z^{\mathbf{h}}\right]_{i,j,k})\right)^2$
Gradient magnitude	$\left|D^{2,\mathbf{h}}(\left[z^{\mathbf{h}}\right]_{i,j,k})\right| := \sqrt{D_{x_1}^{2,\mathbf{h}}(\left[z^{\mathbf{h}}\right]_{i,j,k})+D_{x_2}^{2,\mathbf{h}}(\left[z^{\mathbf{h}}\right]_{i,j,k})+D_{x_3}^{2,\mathbf{h}}(\left[z^{\mathbf{h}}\right]_{i,j,k})}$

$$\partial_{x_2}\left(e(\mathbf{x})\,\partial_{x_3}z(\mathbf{x})\right) \approx D_{x_2}^{\mathbf{h}}\left(\left[e^{\mathbf{h}}\right]_{i,j,k} D_{x_3}^{\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right)\right), \qquad (3.34)$$

$$\partial_{x_3}\left(e(\mathbf{x})\,\partial_{x_1}z(\mathbf{x})\right) \approx D_{x_3}^{\mathbf{h}}\left(\left[d^{\mathbf{h}}\right]_{i,j,k} D_{x_1}^{\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right)\right), \qquad (3.35)$$

$$\partial_{x_3}\left(d(\mathbf{x})\,\partial_{x_2}z(\mathbf{x})\right) \approx D_{x_3}^{\mathbf{h}}\left(\left[e^{\mathbf{h}}\right]_{i,j,k} D_{x_2}^{\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right)\right), \qquad (3.36)$$

$$\partial_{x_3}\left(f(\mathbf{x})\,\partial_{x_3}z(\mathbf{x})\right) \approx D_{x_3}^{-,\mathbf{h}}\left(M_{x_3}^{+,\mathbf{h}}(\left[f^{\mathbf{h}}\right]_{i,j,k}) D_{x_3}^{+,\mathbf{h}}\left(\left[z^{\mathbf{h}}\right]_{i,j,k}\right)\right). \qquad (3.37)$$

The corresponding details on the basic averaging and differential operators can be found in Table 3.2. As in the spatial case the discretisation scheme can be related to a discrete energy functional.

3.3.3 The Discrete Euler–Lagrange Equations

As we have seen in Section 3.2.1, there are basically three types of Euler–Lagrange equations: *linear* ones, *partially nonlinear* ones and *fully nonlinear* ones. Let us now discuss their discretisation by means of the following abbreviations

$$\mathbf{u}_l^{\mathbf{h}} := \left(\left[u_l^{\mathbf{h}} \right]_{1,1}, \ldots, \left[u_l^{\mathbf{h}} \right]_{N_{x_1}^{\mathbf{h}}, N_{x_2}^{\mathbf{h}}} \right)^{\top} \qquad (l = 1, 2), \qquad (3.38)$$

$$\mathbf{j}_{mn}^{\mathbf{h}} := \left(\left[J_{mn} \right]_{1,1}^{\mathbf{h}}, \ldots, \left[J_{mn} \right]_{N_{x_1}^{\mathbf{h}}, N_{x_2}^{\mathbf{h}}}^{\mathbf{h}} \right)^{\top} \qquad (m, n = 1, 2, 3), \qquad (3.39)$$

$$J_{mn}^{\mathbf{h}} := \operatorname{diag}\left(\mathbf{j}_{mn}^{\mathbf{h}\top} \right) \qquad (m, n = 1, 2, 3), \qquad (3.40)$$

and their spatiotemporal counterparts (required for the prototype G). Apart from a compact notation these abbreviations allow us to formulate the resulting systems of equations in such a way that their underlying structure becomes explicit.

3.3.3.1 The Linear Case - Prototypes A-C

Structure. We are now in the position to write down the discrete Euler-Lagrange equations for the linear case. They are given by the following system of equations

$$\left(\underbrace{\begin{pmatrix} J_{11}^{\mathbf{h}} & J_{12}^{\mathbf{h}} \\ J_{12}^{\mathbf{h}} & J_{22}^{\mathbf{h}} \end{pmatrix}}_{\text{linear operator}} - \alpha \underbrace{\begin{pmatrix} L_{\mathrm{L}}^{\mathbf{h}} & 0 \\ 0 & L_{\mathrm{L}}^{\mathbf{h}} \end{pmatrix}}_{\text{linear operator}} \right) \begin{pmatrix} \mathbf{u}_1^{\mathbf{h}} \\ \mathbf{u}_2^{\mathbf{h}} \end{pmatrix} = \underbrace{\begin{pmatrix} -\mathbf{j}_{13}^{\mathbf{h}} \\ -\mathbf{j}_{23}^{\mathbf{h}} \end{pmatrix}}_{\text{rhs}} \qquad (3.41)$$

which is *linear* with respect to the $2N$ unknowns $\mathbf{u}_1^{\mathbf{h}}$ and $\mathbf{u}_2^{\mathbf{h}}$. Here, $L_{\mathrm{L}}^{\mathbf{h}}$ denotes the discrete version of the corresponding linear operator $\mathcal{L}_{\mathrm{L}}$ which comes down to a *symmetric positive semidefinite* $N^{\mathbf{h}} \times N^{\mathbf{h}}$ matrix that has either *pentadiagonal* (prototypes A-B) or *nonadiagonal* (prototype C) structure. These type matrices that actually discretise a diffusion process have been well researched in [Wei94a]. One should note that symmetry and positive semidefiniteness are also shared by the matrix obtained from the data term (by construction) and thus by the overall $2N^{\mathbf{h}} \times 2N^{\mathbf{h}}$ system matrix. One can even show that this matrix is *positive definite* if at least one entry of the off-block diagonals $J_{12}^{\mathbf{h}}$ is different from zero. This means in turn if constant images are excluded – such images are not of interest anyway – the resulting matrix is invertible and the system can be solved by standard numerical techniques (see Chapter 4).

Coupling. One should furthermore note that there are two different types of coupling in the equation system. The *pointwise coupling* between $\mathbf{u}_1^{\mathbf{h}}$ and $\mathbf{u}_2^{\mathbf{h}}$ via the off-block diagonals $J_{12}^{\mathbf{h}}$ in the first operator (data term) and the *neighbourhood coupling* within the main-block diagonals via the second operator $L_{\mathrm{L}}^{\mathbf{h}}$ (smoothness term).

3.3.3.2 The Nonlinear Case I - Prototypes D-E

Structure. Analogously to the linear case, we discretise the Euler Lagrange equations for the first nonlinear case. The obtained nonlinear system of equations reads

$$\left(\underbrace{\begin{pmatrix} J_{11}^{\mathbf{h}} & J_{12}^{\mathbf{h}} \\ J_{12}^{\mathbf{h}} & J_{22}^{\mathbf{h}} \end{pmatrix}}_{\text{linear operator}} - \alpha \underbrace{\begin{pmatrix} L_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}}) & 0 \\ 0 & L_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}}) \end{pmatrix}}_{\text{nonlinear operator}} \right) \begin{pmatrix} \mathbf{u}_1^{\mathbf{h}} \\ \mathbf{u}_2^{\mathbf{h}} \end{pmatrix} = \underbrace{\begin{pmatrix} -\mathbf{j}_{13}^{\mathbf{h}} \\ -\mathbf{j}_{23}^{\mathbf{h}} \end{pmatrix}}_{\text{rhs}}. \qquad (3.42)$$

Here, the finite difference approximation of $\mathcal{L}_{\mathrm{NL}}(u_1, u_2)$ and $\mathcal{L}_{\mathrm{NL}}(u_2, u_1)$ results in the product of a *common nonlinear* operator $L_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}})$ and the vectors $\mathbf{u}_1^{\mathbf{h}}$ and $\mathbf{u}_2^{\mathbf{h}}$, respectively. One should note that for a fixed argument this nonlinear argument becomes linear and the solution process comes down to the linear case. This aspect will be exploited in the next chapter.

Coupling. Apart from the point coupling and the neighbourhood coupling that are already present in the linear case, a third type of coupling of the unknowns is present. It is given by the joint argument of the common nonlinear operator $L_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}})$.

3.3.3.3   The Nonlinear Case II - Prototypes F-G

Structure. As in the previous case the discretisation of the Euler-Lagrange equations for the nonlinear case II yields a nonlinear system of equations. It is given by

$$
\left(
\underbrace{
\begin{pmatrix}
F_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}})J_{11}^{\mathbf{h}} & F_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}})J_{12}^{\mathbf{h}} \\
F_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}})J_{12}^{\mathbf{h}} & F_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}})J_{22}^{\mathbf{h}}
\end{pmatrix}
}_{\text{nonlinear operator}}
- \alpha
\underbrace{
\begin{pmatrix}
L_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}}) & 0 \\
0 & L_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}})
\end{pmatrix}
}_{\text{nonlinear operator}}
\right)
\begin{pmatrix}
\mathbf{u}_1^{\mathbf{h}} \\
\mathbf{u}_2^{\mathbf{h}}
\end{pmatrix}
$$

$$
+ \underbrace{
\begin{pmatrix}
F_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}})\, \mathbf{j}_{13}^{\mathbf{h}} \\
F_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}})\, \mathbf{j}_{23}^{\mathbf{h}}
\end{pmatrix}
}_{\text{nonlinear operator}}
= \underbrace{
\begin{pmatrix}
\mathbf{0} \\
\mathbf{0}
\end{pmatrix}
}_{\text{rhs}}
\tag{3.43}
$$

where the nonlinear factor $\Psi_{\mathrm{D}}'(\mathbf{u}^\top J\, \mathbf{u})$ results in the nonlinear operator

$$
F_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}}) := \mathrm{diag}\Big(\Psi_{\mathrm{D}}'\big([\mathbf{u}^{\mathbf{h}}]_{1,1}^\top [J^{\mathbf{h}}]_{1,1} [\mathbf{u}^{\mathbf{h}}]_{1,1} \big), \ldots, \tag{3.44}
$$
$$
\ldots, \Psi_{\mathrm{D}}'\big([\mathbf{u}^{\mathbf{h}}]_{N_{x_1}^{\mathbf{h}}, N_{x_2}^{\mathbf{h}}}^\top [J^{\mathbf{h}}]_{N_{x_1}^{\mathbf{h}}, N_{x_2}^{\mathbf{h}}} [\mathbf{u}^{\mathbf{h}}]_{N_{x_1}^{\mathbf{h}}, N_{x_2}^{\mathbf{h}}} \big) \Big).
$$

One should note that in contrast to the nonlinear case I, the right hand side is homogeneous. This is a direct consequence of the fact that all terms are nonlinear (they can be combined into *one single nonlinear operator*).

Coupling. One should note that the linear point coupling in the data term that appears in the first two cases (prototypes A-E) may now become strongly nonlinear, since it is re-weighted by the entries of the nonlinear operator $F_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}})$. The use of a joint argument for this operators thereby constitutes a fourth type of coupling.

3.4   Qualitative Experiments

After we have seen how to minimise the obtained Euler–Lagrange equations – efficient strategies for this purpose will be discussed in the next chapter – let us now perform a systematic comparison of all prototypes with respect to their estimation quality. To this end, we evaluate the performance of the different approaches for various motion scenarios: We start with sequences based on *global motion types*, continue with *moving objects* that require the estimation of *discontinuities*, and end up with scenes where *both camera and objects are moving*. Moreover, we consider image sequences with *varying illumination*

and *noise*. In order to give a realistic impression of the true capabilities of each prototype both synthetic and real-world sequences are used. This also allows to address the problem of *interlacing* that is typical for many applications.

3.4.1 Synthetic Sequences

In contrast to real-world sequences, synthetic sequences offer the advantage that they allow to measure the performance of an approach quantitatively. As explained in Chapter 2, this is done by means of the average angular error (AAE) [BFB94]. In the following we present six experiments that investigate the performance of the proposed prototypes for synthetic sequences in both a quantitative and a qualitative way.

Tab. 3.3: **Experiment I**: Performance of the prototypes for synthetic sequences with globally divergent motion. We optimised the parameters σ_{spat}, σ_{temp}, α, ρ and γ with respect to the average angular error (AAE).

Yosemite Sequence without Clouds

prototype	method	σ_{spat}	α	other parameters	AAE
A	homogeneous	1.80	800	-	2.64°
B	image-driven isotropic	1.80	2700	-	2.64°
C	image-driven anisotropic	1.80	3000	-	2.57°
D	flow-driven isotropic	1.80	27	-	2.58°
E	flow-driven anisotropic	1.80	22	-	2.65°
F	Bruhn *et al.* 2-D	1.95	17	$\rho=1.3$	2.27°
G	Papenberg *et al.* 3-D	1.95	7	$\gamma=1.0$, $\sigma_{\mathrm{temp}}=0.80$	1.45°

3.4.1.1 Experiment I: Globally Divergent Motion

In our first experiment we study the performance of the seven prototypes with respect to globally divergent motion. To this end, we used the *Yosemite* sequence *without* sky by Black (`http://www.cs.brown.edu/people/black/images.html`). It consists of 15 frames of size 316×252 and depicts a flight through the Yosemite National Park. The occurring displacements are in the order up to 5.5 pixels. In order to allow for a more accurate evaluation of the results we did not consider the 8-bit ground truth provided with the image data, but the original 16-bit ground truth [1] by John Barron. Moreover, we omitted the sky region when computing the average angular error. This is common practice, since the sky does not provide any texture. One should note that this simplifies the estimation significantly, since there are no real discontinuities left.

The obtained average angular errors for all prototypes are listed in Table 3.3. As one would expect for such a simple task, all methods perform relatively good. However, one can see a clear difference between the results of the basic prototypes A-E and the results of the advanced methods F-G. This is not surprising, since due to the absence of discontinuities other concepts such as robust data terms (F,G), local least square fits (F)

---

1. In the scope of the performance benchmark of Barron *et al.* [BFB94] this ground truth was available at `ftp://csd.uwo.ca/pub/vision`.

Fig. 3.1: **Experiment I**: Performance of the prototypes for synthetic sequences with globally divergent motion. **(a) Top Left:** Frame 8 of the *Yosemite* sequence *without* clouds of size 316×256 (grey-scale). **(b) Top Centre:** Frame 9. **(c) Top Right:** Ground truth (vector plot). **(d) Upper Middle Left:** Ground truth (colour plot). **(e) Upper Middle Centre:** Prototype A (homogeneous regularisation). **(f) Upper Middle Right:** Prototype B (image-driven isotropic regularisation). **(g) Lower Middle Left:** Prototype C (image-driven anisotropic regularisation). **(h) Lower Middle Centre:** Prototype D (flow-driven isotropic regularisation). **(i) Lower Middle Right:** Prototype E (flow-driven anisotropic regularisation). **(j) Bottom Left:** Prototype F (Bruhn *et al.* 2-D, SD). **(k) Bottom Centre:** Prototype G (Papenberg *et al.* 3-D, SD).

Fig. 3.2: **Experiment II**: Performance of the prototypes for synthetic sequences with discontinuities, outliers and globally divergent motion. **(a) Top Left:** Frame 10 of the *Office* sequence of size 200×200 (grey-scale variant). **(b) Top Centre:** Frame 11. **(c) Top Right:** Ground truth (vector plot). **(d) Upper Middle Left:** Ground truth (colour plot). **(e) Upper Middle Centre:** Prototype A (homogeneous regularisation). **(f) Upper Middle Right:** Prototype B (image-driven isotropic regularisation). **(g) Lower Middle Left:** Prototype C (image-driven anisotropic regularisation). **(h) Lower Middle Centre:** Prototype D (flow-driven isotropic regularisation). **(i) Lower Middle Right:** Prototype E (flow-driven anisotropic regularisation). **(j) Bottom Left:** Prototype F (Bruhn *et al.* 2-D, SD). **(k) Bottom Centre:** Prototype G (Papenberg *et al.* 3-D, SD).

and spatiotemporal regularisers (G) become more important. In fact, one can see from the corresponding flow fields in Figure 3.1 that the advanced methods allow for a much smoother estimation of the results. Outliers at the lower left corner are thereby eliminated successfully by the robust data term and the additional temporal smoothness.

3.4.1.2   Experiment II: Discontinuities, Outliers, Globally Divergent Motion

In our second experiment we investigate the performance of the different prototypes with respect to global divergent motion, outliers and discontinuities. For this purpose we considered the *Office* sequence by Galvin *et al.* [GMN+98]. This sequence consists of 20 frames of size 200×200 and depicts a zoom into a synthetic working place. The varying depth of the furniture thereby creates small discontinuities, that have to be estimated correctly by the different approaches. However, the displacements are rather small, since they do not exceed 1.5 pixels. Apart from with the discontinuities, the methods have also to deal with temporal aliasing artifacts that manifest themselves in a large number of outliers in regions of the monitor, the plant and the window.

Let us take a look at the achieved results that are presented in Table 3.4. They show that apart from the advanced techniques F and G, also the simple prototype A performs well – although this approach does not respect motion discontinuities. While this may be surprising at first glance, it has a simple explanation: If we compare the corresponding flow fields depicted in Figure 3.2, we see that the non-robust discontinuity preserving methods B-E not only respect edges in the image and flow field: They also adapt to outliers that have been mentioned before. Thus, although their performance is superior in most areas, they cannot outperform the evidently too smooth result of the prototype A [2]. This also explains the good performance of the advanced techniques F and G. Here, the robust data term and the spatiotemporal regularisation contribute significantly to the improved results.

Tab. 3.4: **Experiment II**: Performance of the prototypes for synthetic sequences with discontinuities, outliers and globally divergent motion. We optimised the parameters σ_{spat}, σ_{temp}, α, ρ and γ with respect to the average angular error (AAE).

<div align="center">

Office Sequence

prototype	method	σ_{spat}	α	other parameters	AAE
A	homogeneous	0.75	2700	-	4.36°
B	image-driven isotropic	0.75	5400	-	5.06°
C	image-driven anisotropic	0.60	10000	-	5.03°
D	flow-driven isotropic	0.75	70	-	4.99°
E	flow-driven anisotropic	0.75	49	-	4.94°
F	Bruhn *et al.* 2-D, SD	0.80	17	ρ=2.5	4.31°
G	Papenberg *et al.* 3-D, SD	1.30	7	γ=2.4, σ_{temp}=0.10	3.26°

</div>

---

2. Please note that we have kept the regularisation parameter ϵ_{S} for each prototype fixed in all our experiments. Increasing its value to a large number, however, would allow to approximate the behaviour of prototype A (homogeneous regularisation) with the prototypes B-E (adaptive regularisation) and thus to achieve similar results for all five basic prototypes.

3.4.1.3 Experiment III: Discontinuities, Translational Motion with Static Background

After we have investigated the performance of the different prototypes with respect to divergent motion, let us now study their capabilities of estimating translational motion. This is done in our third experiment by the example of Middendorf's *New Marble* sequence (`http://i21www.ira.uka.de/image_sequences/`). This sequence was already used in our theoretical chapter and consists of 200 frames of size 512×384. Apart from a static marble block it depicts two moving blocks that slide over a marble floor with up to 2 pixels per frame. Due to the static background this scene requires the estimation of sharp motion discontinuities at the boundaries of all moving objects.

The corresponding results are presented in Table 3.5. As one can see all prototypes allow for an accurate estimation of the results. However, the raw numbers also show the advantage of flow-driven methods for heavily textured sequences. In contrast to image-driven techniques they allow for a homogeneous estimation of moving objects even across discontinuities that belong to the texture while still respecting true motion discontinuities. This is also confirmed by the computed flow fields that are depicted in Figure 3.3. They show the best results for the methods D-G. Thereby the advantage of robust data terms becomes obvious: Only the prototypes F and G allow to suppress the motion that results from the shadow of the large block moving to the right.

Tab. 3.5: **Experiment III**: Performance of the prototypes for synthetic sequences with discontinuities and translational motion with static background. We optimised the parameters σ_{spat}, σ_{temp}, α, ρ and γ with respect to the average angular error(AAE).

New Marble Sequence

prototype	method	σ_{spat}	α	other parameters	AAE
A	homogeneous	0.60	850	-	2.51°
B	image-driven isotropic	0.60	3050	-	2.45°
C	image-driven anisotropic	0.60	3050	-	2.51°
D	flow-driven isotropic	0.60	47	-	1.50°
E	flow-driven anisotropic	0.60	38	-	1.58°
F	Bruhn *et al.* 2-D, SD	0.50	27	ρ=0.1	0.93°
G	Papenberg *et al.* 3-D, SD	0.50	19	γ=1.0, σ_{temp}=0.10	0.66°

3.4.1.4 Experiment IV: Discontinuities, Outliers, Translational Motion with Moving Background

In the previous experiment, objects were moving in front of a *static background*. In order to make things more challenging, let us evaluate in our fourth experiment the performance of out prototypes if *both* object and background are moving. To this end, we consider the *Street* sequence by Galvin *et al.* [GMN$^+$98]. This synthetic sequence consists of 20 frames of size 200×200 and depicts a car driving from the left to the right with up to 4.5 pixels per frame. At the same time – induced by a slow camera motion following the car – the background moves in the opposite direction. Evidently, this scenario requires the estimation of two types of translational motion as well as the preservation of motion discontinuities. Moreover, it requires the handling of a significant number of data outliers
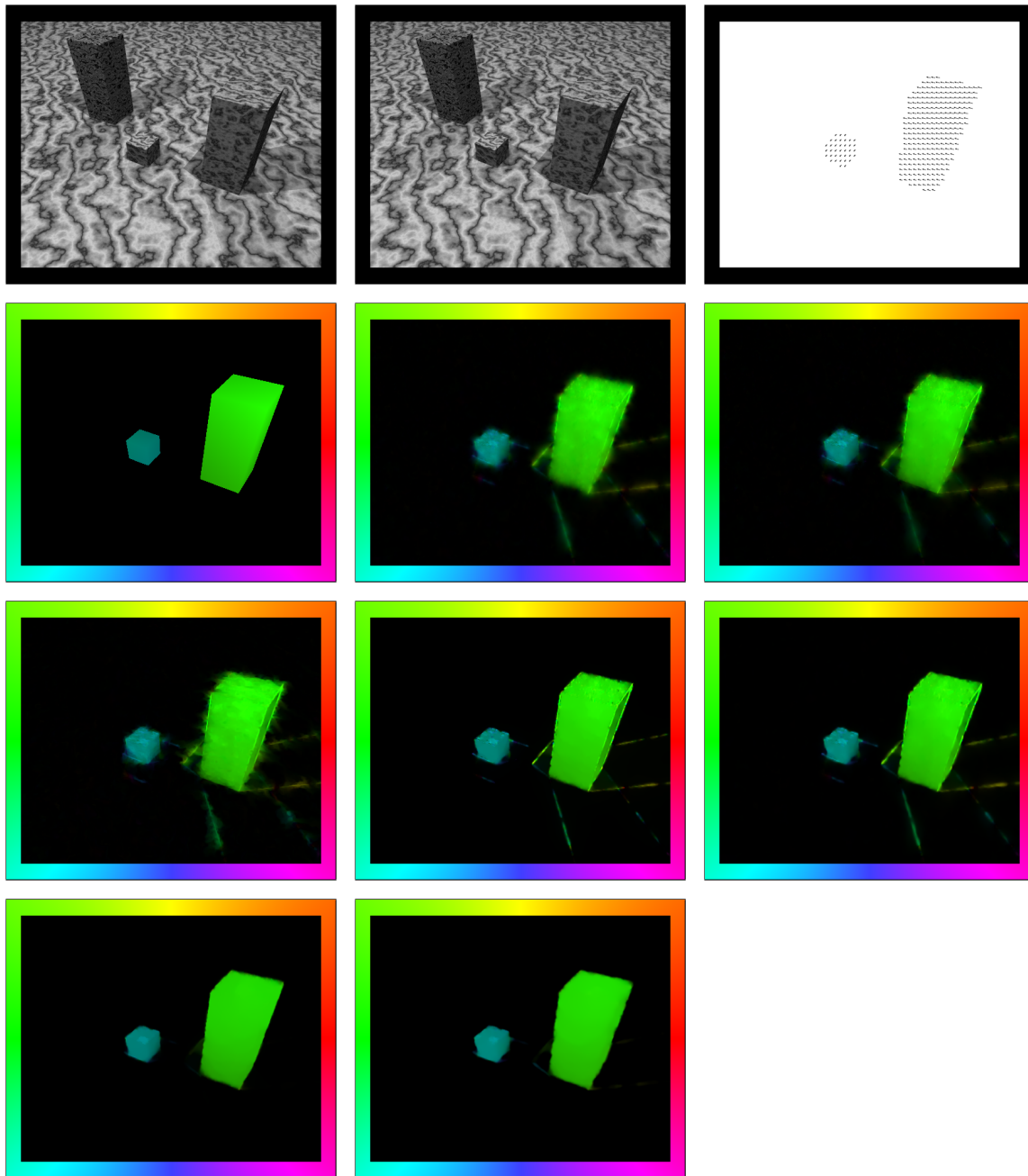
Fig. 3.3:   **Experiment III**: Performance of the prototypes for synthetic sequences with discontinuities, outliers and translational motion with static background. **(a) Top Left:** Frame 150 of the *New Marble* sequence of size 512×384 (grey-scale variant). **(b) Top Centre:** Frame 151. **(c) Top Right:** Ground truth (vector plot). **(d) Upper Middle Left:** Ground truth (colour plot). **(e) Upper Middle Centre:** Prototype A (homogeneous regularisation). **(f) Upper Middle Right:** Prototype B (image-driven isotropic regularisation). **(g) Lower Middle Left:** Prototype C (image-driven anisotropic regularisation). **(h) Lower Middle Centre:** Prototype D (flow-driven isotropic regularisation). **(i) Lower Middle Right:** Prototype E (flow-driven anisotropic regularisation). **(j) Bottom Left:** Prototype F (Bruhn *et al.* 2-D, SD). **(k) Bottom Centre:** Prototype G (Papenberg *et al.* 3-D, SD).

Fig. 3.4: **Experiment IV**: Performance of the prototypes for synthetic sequences with discontinuities, outliers and translational motion with dynamic background (camera motion). **(a) Top Left:** Frame 10 of the *Street* sequence of size 200×200 (grey-scale variant). **(b) Top Centre:** Frame 11. **(c) Top Right:** Ground truth (vector plot). **(d) Upper Middle Left:** Ground truth (colour plot). **(e) Upper Middle Centre:** Prototype A (homogeneous regularisation). **(f) Upper Middle Right:** Prototype B (image-driven isotropic regularisation). **(g) Lower Middle Left:** Prototype C (image-driven anisotropic regularisation). **(h) Lower Middle Centre:** Prototype D (flow-driven isotropic regularisation). **(i) Lower Middle Right:** Prototype E (flow-driven anisotropic regularisation). **(j) Bottom Left:** Prototype F (Bruhn *et al.* 2-D, SD). **(k) Bottom Centre:** Prototype G (Papenberg *et al.* 3-D, SD).

(e.g. in the region of the tree). These outliers are once again the result of temporal aliasing and manifest themselves in flickering artifacts.

The obtained average angular errors for all our methods are listed in Table 3.6. As one can see, the image-driven prototypes A-C perform relatively poor. This can be explained as follows: Since these methods either favour very smooth solutions (prototype A) or adapt to the (false) image data (prototype B-C), they are not suitable for corrupted image sequences with small moving objects. This is also reflected in the fact that we had to *limit* the value of the smoothness parameter α to 2000 for these three prototypes in order to prevent them from ignoring the foreground object and estimating only the translational background motion. In contrast, the flow-driven prototypes D-G allow to preserve discontinuities without adapting directly to the corrupted data. Therefore it is not surprising that they give significantly better results. This is also confirmed by the corresponding flow fields depicted in Figure 3.4. As one can see the background is estimated much better in the cases D-G. Thereby it becomes also obvious why the advanced prototype F is outperformed by the basic prototypes D-E. While its robust data term allows an even better estimation of the background it also treats parts of the (relatively small) car as outliers. This, of course, reduces the overall accuracy. As the flow field of prototype G shows, spatiotemporal information may be useful to overcome this problem. In fact, for this prototype, both types of motion are estimated accurately.

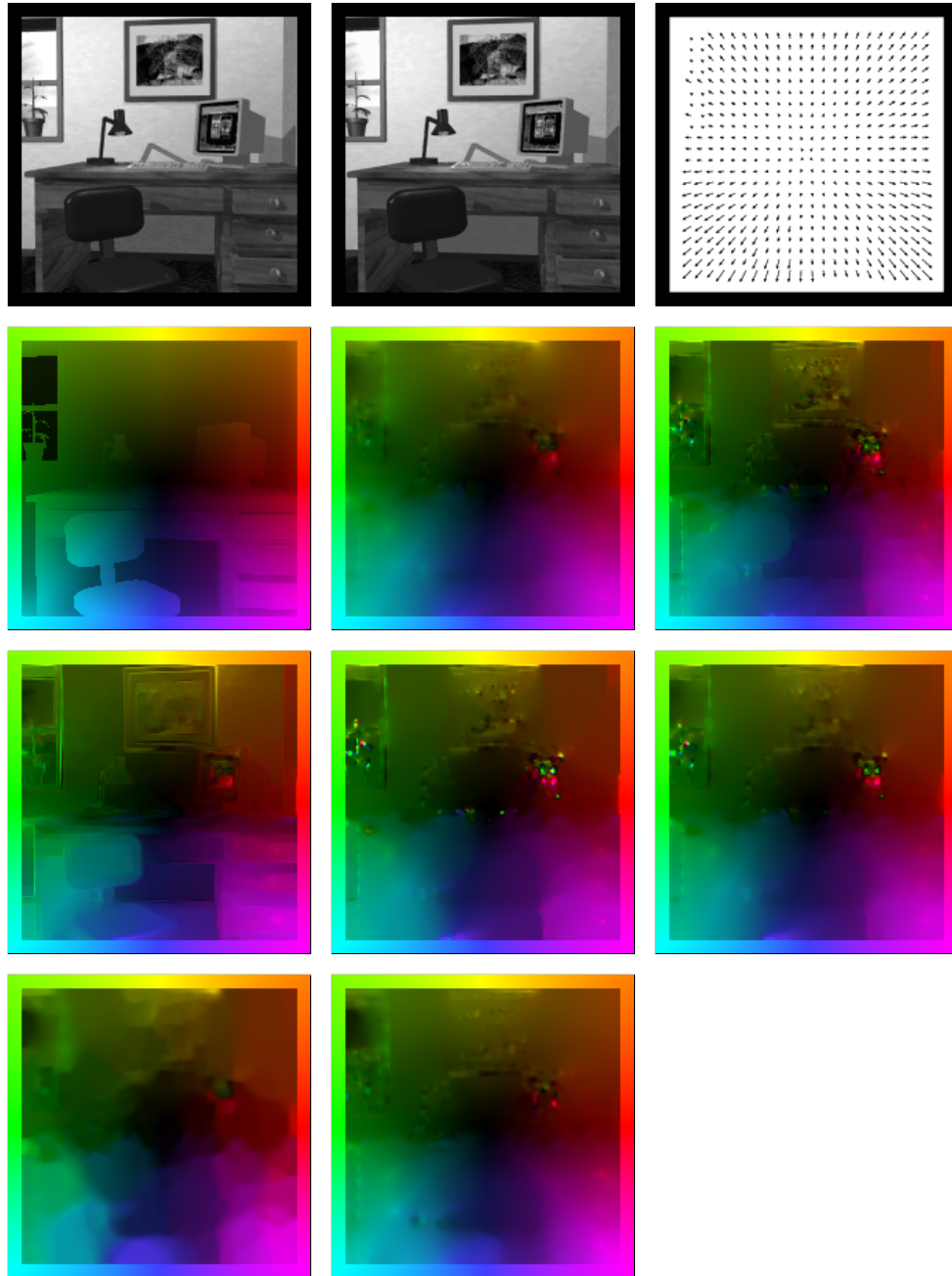**Tab. 3.6**: **Experiment IV**: Performance of the prototypes for synthetic sequences with discontinuities and translational motion with dynamic background camera motion. We optimised the parameters σ_{spat}, σ_{temp}, α, ρ and γ with respect to the average angular error (AAE).

Street Sequence

prototype	method	σ_{spat}	α	other parameters	AAE
A	homogeneous	0.60	2000	-	6.40°
B	image-driven isotropic	0.70	2000	-	6.48°
C	image-driven anisotropic	0.60	2000	-	5.79°
D	flow-driven isotropic	0.80	90	-	4.09°
E	flow-driven anisotropic	0.80	65	-	4.13°
F	Bruhn *et al.* 2-D, SD	0.90	10	ρ=0.9	4.30°
G	Papenberg *et al.* 3-D, SD	1.60	11	γ=2.1, σ_{temp}=0.10	3.21°

3.4.1.5 Experiment V: Discontinuities, Translational and Divergent Motion, Varying Illumination

Until now we have performed experiments for sequences with *either* translational *or* divergent motion. In our fifth experiment, however, we evaluate the performance for sequences where *both types of motions* are present. To this end, once more the famous *Yosemite* sequence *with* clouds is used. This sequence consists of 15 frames of size 316×252 and depicts a flight through the Yosemite national park. It combines the divergent motion of the flight with the translational motion the clouds. The transition between both motion types thereby requires the estimation of a sharp discontinuity. Moreover, the region of the clouds is very challenging, since varying illumination is present.

The corresponding results for this sequence are listed in Table 3.7. They nicely show how additional concepts can improve the estimation: One can see that non-adaptive reg-

ularisers (prototype A) are outperformed by adaptive ones (prototype B-G), that robust data terms (prototypes F-G) give better results than non-robust ones (prototypes A-E) and that the gradient constancy assumption (prototype G) is much more suitable under varying illumination than the grey value constancy assumption (A-F). In fact, all these observations are confirmed by the corresponding fields depicted in Figure 3.7. The discontinuity-preserving methods allow the estimation of a sharp discontinuity at the horizon, the robust methods overcome the small outlier problem at the lower left corner (these pixels move out of the image) and the prototype based on the gradient constancy assumption allows the correct estimation of the sky region despite of the varying illumination. One can also see that all methods underestimate the optic flow vectors in the lower left corner (the cyan is not as bright as the one of the ground truth). This, however, is not surprising: Since the correct displacements are in the order of seven pixels in this area, our methods based on linearised data terms have slight problems. Nevertheless, the estimation is still pretty accurate as the average angular errors show.

Tab. 3.7: **Experiment V**: Performance of the prototypes for synthetic sequences with discontinuities, translational and divergent motion as well as varying illumination. We optimised the parameters σ_{spat}, σ_{temp}, α, ρ and γ with respect to the average angular error (AAE).

<div align="center">

The Yosemite Sequence with Clouds

prototype	method	σ_{spat}	α	other parameters	AAE
A	homogeneous	1.30	500	-	7.17°
B	image-driven isotropic	1.20	2700	-	6.44°
C	image-driven anisotropic	1.30	4500	-	6.28°
D	flow-driven isotropic	1.30	42	-	6.32°
E	flow-driven anisotropic	1.30	44	-	6.42°
F	Bruhn *et al.* 2-D, SD	1.60	24	ρ=1.4	5.74°
G	Papenberg *et al.* 3-D, SD	2.10	93	γ=1050, σ_{temp}=0.50	2.78°

</div>

3.4.1.6 Experiment VI: Discontinuities, Translational and Divergent Motion, Varying Illumination, Gaussian Noise

Our sixth and last experiment with respect to the synthetic sequences is dedicated to the performance of the different prototypes under noise. To this end, we added Gaussian noise of standard deviation $\sigma_n = 10, 20$ and 40 to the *Yosemite* sequence *with* clouds and computed the average angular error for all prototypes.

The obtained results for all methods and all noise levels are presented in Table 3.8. As expected, the advanced prototype F yields the most accurate flow fields of all *spatial* prototypes A-F. However, compared to the *spatiotemporal* prototypes G its performance looks rather poor. This has two reasons: Firstly, the gradient constancy assumption is by far the most suitable constancy assumption for this sequence (due to the varying illumination in the sky). This is already reflected in the results without noise. Secondly, the robust data term and the additional temporal information compensate for the higher sensitivity of this constancy assumption with respect to noise. Thus it is not surprising that average angular errors are obtained that are almost twice as low as those of the remaining prototypes. Let us now take a look at the computed flow fields for the noise level of $\sigma_n = 40$ (Figure

Fig. 3.5: **Experiment V**: Performance of the prototypes for synthetic sequences with discontinuities, translational and divergent motion as well as varying illumination. **(a) Top Left:** Frame 8 of the *Yosemite* sequence *with* clouds of size 316×256 (grey-scale). **(b) Top Centre:** Frame 9. **(c) Top Right:** Ground truth (vector plot). **(d) Upper Middle Left:** Ground truth (colour plot). **(e) Upper Middle Centre:** Prototype A (homogeneous regularisation). **(f) Upper Middle Right:** Prototype B (image-driven isotropic regularisation). **(g) Lower Middle Left:** Prototype C (image-driven anisotropic regularisation). **(h) Lower Middle Centre:** Prototype D (flow-driven isotropic regularisation). **(i) Lower Middle Right:** Prototype E (flow-driven anisotropic regularisation). **(j) Bottom Left:** Prototype F (Bruhn *et al.* 2-D, SD). **(k) Bottom Centre:** Prototype G (Papenberg *et al.* 3-D, SD).

Fig. 3.6: **Experiment VI:** Performance of the prototypes for synthetic sequences with Gaussian noise. **(a) Top Left:** Frame 8 of the *Yosemite* sequence *with* clouds of size 316×256 (grey-scale). Gaussian noise of $\sigma_n = 40$ was added. **(b) Top Centre:** Frame 9. **(c) Top Right:** Ground truth (vector plot). **(d) Upper Middle Left:** Ground truth (colour plot). **(e) Upper Middle Centre:** Prototype A (homogeneous regularisation). **(f) Upper Middle Right:** Prototype B (image-driven isotropic regularisation). **(g) Lower Middle Left:** Prototype C (image-driven anisotropic regularisation). **(h) Lower Middle Centre:** Prototype D (flow-driven isotropic regularisation). **(i) Lower Middle Right:** Prototype E (flow-driven anisotropic regularisation). **(j) Bottom Left:** Prototype F (Bruhn *et al.* 2-D, SD). **(k) Bottom Centre:** Prototype G (Papenberg *et al.* 3-D, SD).
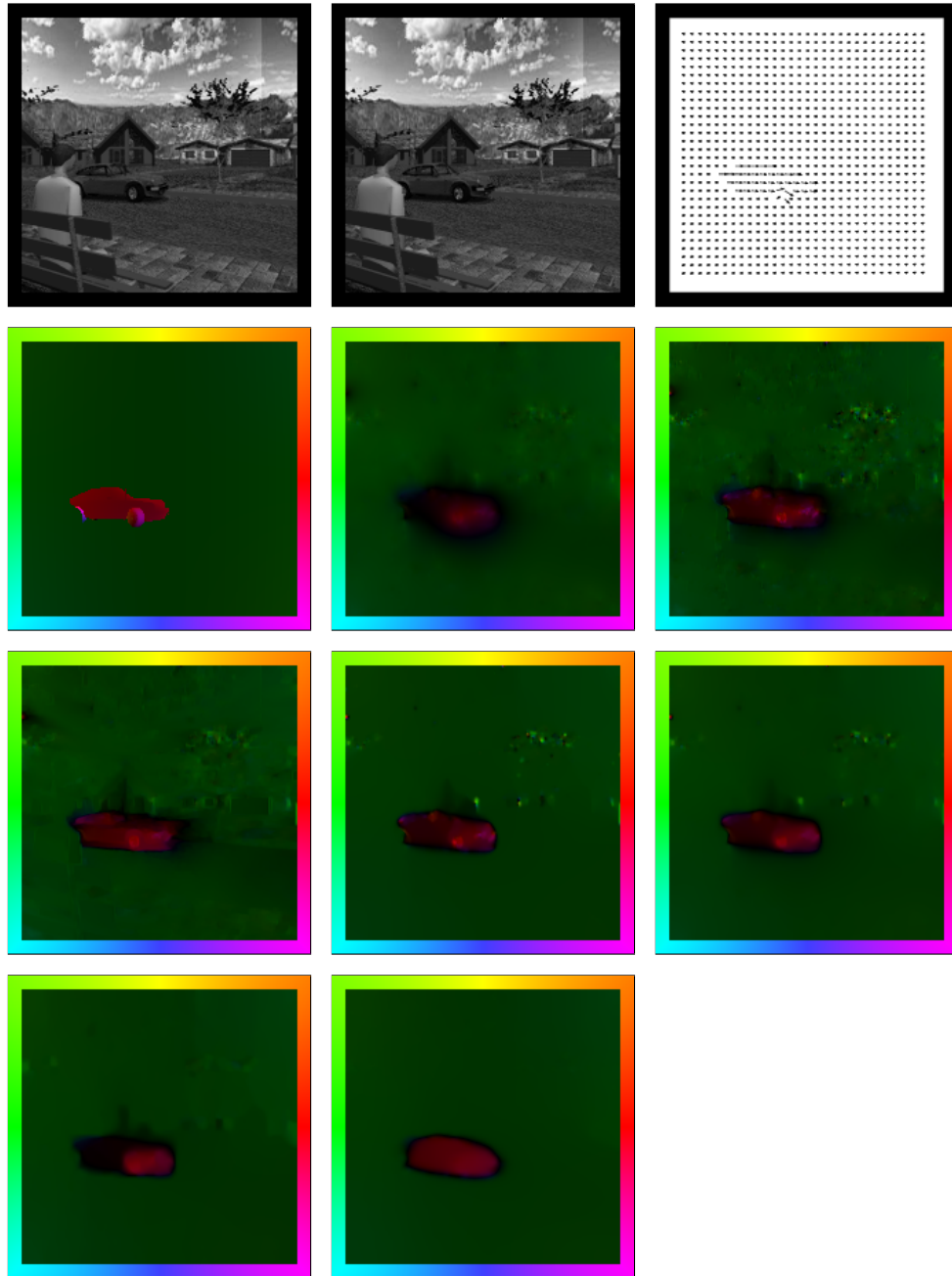
Tab. 3.8: **Experiment VI:** Performance of the prototypes for synthetic sequences with Gaussian noise. We optimised the parameters σ_{spat}, σ_{temp}, α, ρ and γ with respect to the average angular error (AAE).

Yosemite Sequence with Clouds - Noise Level $\sigma_n = 0$

prototype	method	σ_{spat}	α	other parameters	AAE
A	homogeneous	1.30	500	-	7.17°
B	image-driven isotropic	1.20	2700	-	6.44°
C	image-driven anisotropic	1.30	4500	-	6.28°
D	flow-driven isotropic	1.30	42	-	6.32°
E	flow-driven anisotropic	1.30	44	-	6.42°
F	Bruhn *et al.* 2-D, SD	1.60	24	ρ=1.4	5.74°
G	Papenberg *et al.* 3-D, SD	2.10	60	γ=1050, σ_{temp}=0.50	2.78°

Yosemite Sequence with Clouds - Noise Level $\sigma_n = 10$

prototype	method	σ_{spat}	α	other parameters	AAE
A	homogeneous	1.80	1100	-	9.37°
B	image-driven isotropic	1.80	4500	-	8.72°
C	image-driven anisotropic	1.80	6600	-	8.17°
D	flow-driven isotropic	1.80	70	-	8.22°
E	flow-driven anisotropic	1.80	56	-	8.28°
F	Bruhn *et al.* 2-D, SD	1.80	20	ρ=2.3	7.96°
G	Papenberg *et al.* 3-D, SD	3.10	39	γ=1050, σ_{temp}=0.40	3.95°

Yosemite Sequence with Clouds - Noise Level $\sigma_n = 20$

prototype	method	σ_{spat}	α	other parameters	AAE
A	homogeneous	2.10	2000	-	12.17°
B	image-driven isotropic	2.10	7500	-	11.62°
C	image-driven anisotropic	2.10	12500	-	10.99°
D	flow-driven isotropic	2.10	110	-	10.91°
E	flow-driven anisotropic	2.10	90	-	11.03°
F	Bruhn *et al.* 2-D, SD	2.10	21	ρ=3.0	10.71°
G	Papenberg *et al.* 3-D, SD	3.80	36	γ=1050, σ_{temp}=0.65	5.58°

Yosemite Sequence with Clouds - Noise Level $\sigma_n = 40$

prototype	method	σ_{spat}	α	other parameters	AAE
A	homogeneous	2.40	4500	-	16.82°
B	image-driven isotropic	2.40	17000	-	16.37°
C	image-driven anisotropic	2.40	32000	-	15.73°
D	flow-driven isotropic	2.40	191	-	15.66°
E	flow-driven anisotropic	2.40	178	-	15.69°
F	Bruhn *et al.* 2-D, SD	2.40	12	ρ=15.4	15.31°
G	Papenberg *et al.* 3-D, SD	5.00	25	γ=1050, σ_{temp}=0.70	8.29°

3.6). As one can see, they confirm our considerations with respect to the advantages of our prototype G: In spite of the large amount of noise, the gradient constancy assumption still allows for a precise estimation of the sky region.

3.4.2 Real-World Sequences

In the previous six experiments we have evaluated the performance of our prototypes with respect to synthetic sequences. Let us now study by the example of three real-world scenes how these methods actually perform on non-synthetic data. To this end, we start with a simple scene with rotational motion – a motion type that has not been considered to so far.

3.4.2.1 Experiment I: Discontinuities, Rotational Motion

In our first experiment on real-world data we evaluate the performance of our prototypes with respect to rotational motion. To this end, we selected the *Rubik* sequence by Szeliski [3] that consists of 21 frames of size 256×240. This sequence shows Rubik's famous cube positioned on a rotating disk and requires the estimation of discontinuities within a rotational motion. The occurring displacements are approximately in the order of 1.5 pixels.

As common for real-world sequences, the ground truth for the Rubik scene is not known. Thus, we adapted the parameters in such a way that the visual quality was optimised (homogeneous estimation of disc and cube). The corresponding flow fields are depicted in Figure 3.7. Surprisingly, they show an very good performance of our image-driven prototypes B and C. In particular, the flow field obtained by our anisotropic technique C looks very realistic. Almost all flow-driven methods, however, suffer from the problem that they propagate information into the textureless static floor. At this point one should note that this behaviour is *fully in accordance* with the underlying model assumptions: Pixels of the same grey value are still mapped to each other. Only the *prior knowledge* that the floor is solid and thus does not move leads to our decision that this filling-in of flow information is wrong. The only exception from this behaviour is given by our spatiotemporal prototype G. Due to the additional temporal information this prototype requires only a relatively small value for the smoothness weight α. This in turn allows for a homogeneous estimation of the motion for the disc and the cube without smoothing over the corresponding boundaries.

3.4.2.2 Experiment II: Discontinuities, Multiple Translational Motions, Interlacing

So far we have restricted ourselves to image sequences with not more than two moving objects. However, in particular with respect to computer vision tasks such as video surveillance or people tracking, optic flow approaches should allow the simultaneous estimation of multiple motions. This aspect is investigated in our second experiment on real-world data, where a classic traffic sequence is used: the *Rheinhafen* scene by Nagel. This scene that is available from `http://i21www.ira.uka.de/image_sequences/` shows a surveillance video of a highly frequented intersection and consists of 1000 frames of size 688×565. Apart from estimating the motion of the different cars (up to 9 pixels per frame),

---

3. In the scope of the performance benchmark of Barron *et al.* [BFB94] this sequence was available at `ftp://csd.uwo.ca/pub/vision`.
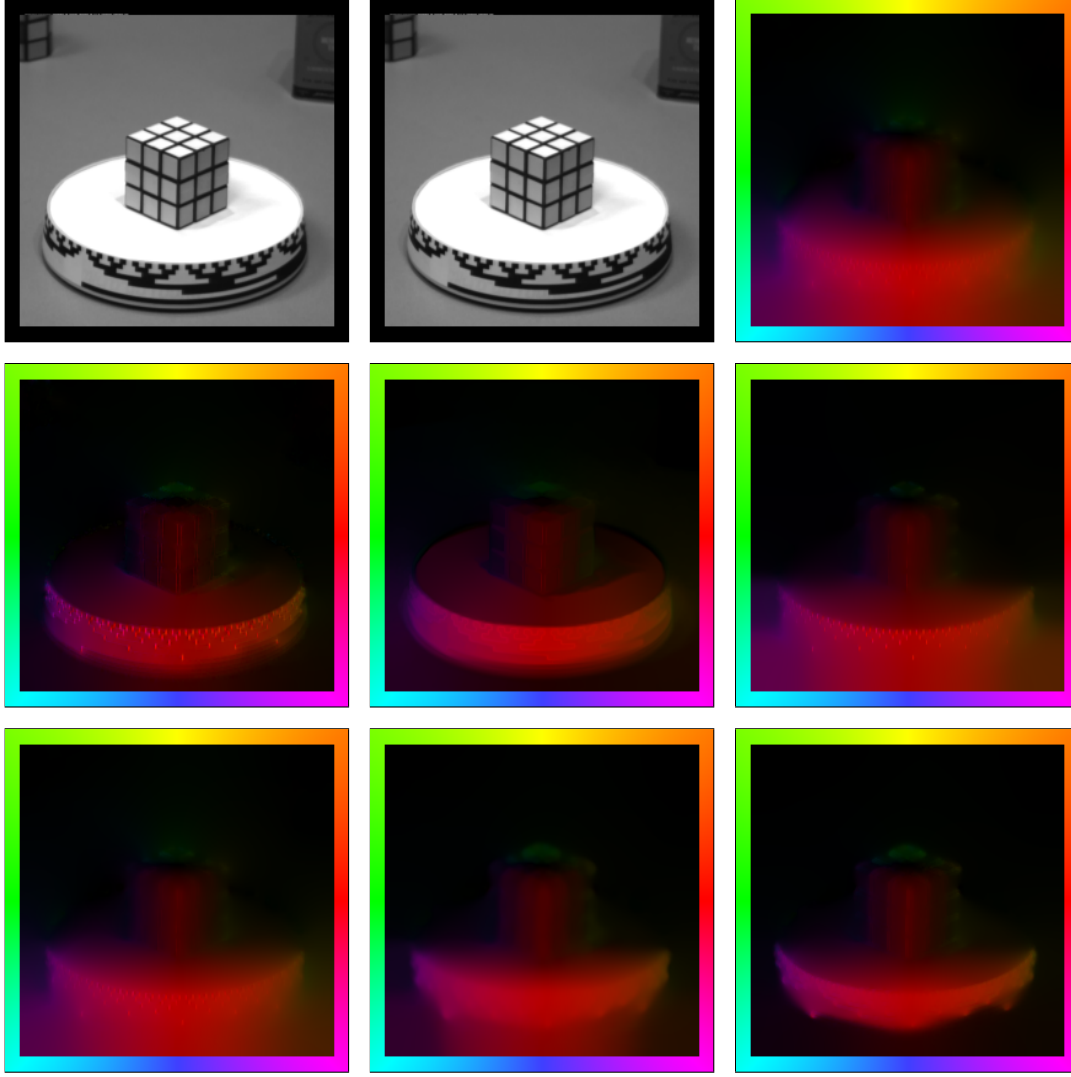
Fig. 3.7: **Experiment I**: Performance of the prototypes for real-world sequences with discontinuities and rotational motion. **(a) Top Left:** Frame 10 of the *Rubic* sequence of size 256×240 (grey-scale). **(b) Top Centre:** Frame 11. **(c) Top Right:** Prototype A (homogeneous regularisation). **(d) Middle Left:** Prototype B (image-driven isotropic regularisation). **(e) Middle Centre:** Prototype C (image-driven anisotropic regularisation). **(f) Middle Right:** Prototype D (flow-driven isotropic regularisation). **(g) Bottom Left:** Prototype E (flow-driven anisotropic regularisation). **(h) Bottom Centre:** Prototype F (Bruhn *et al.* 2-D, SD). **(i) Bottom Right:** Prototype G (Papenberg *et al.* 3-D, SD).

this scene also requires to deal with interlacing artifacts. These stripe artifacts that result from an alternating update of even and odd lines are typical for real-world applications.

The computed flow fields for this sequence are depicted in Figure 3.9. As one can see, the interlacing artifacts hardly spoil the estimation. Only in the case of our image-driven prototypes B and C they yield slightly visible stripe-artifacts (within the black background). In particular for the flow-driven prototypes C-G, however, the estimation quality is very good. The motion boundaries are sharp and even the motion of the small cars in the background is estimated precisely. Nevertheless, one should note that all methods suffer
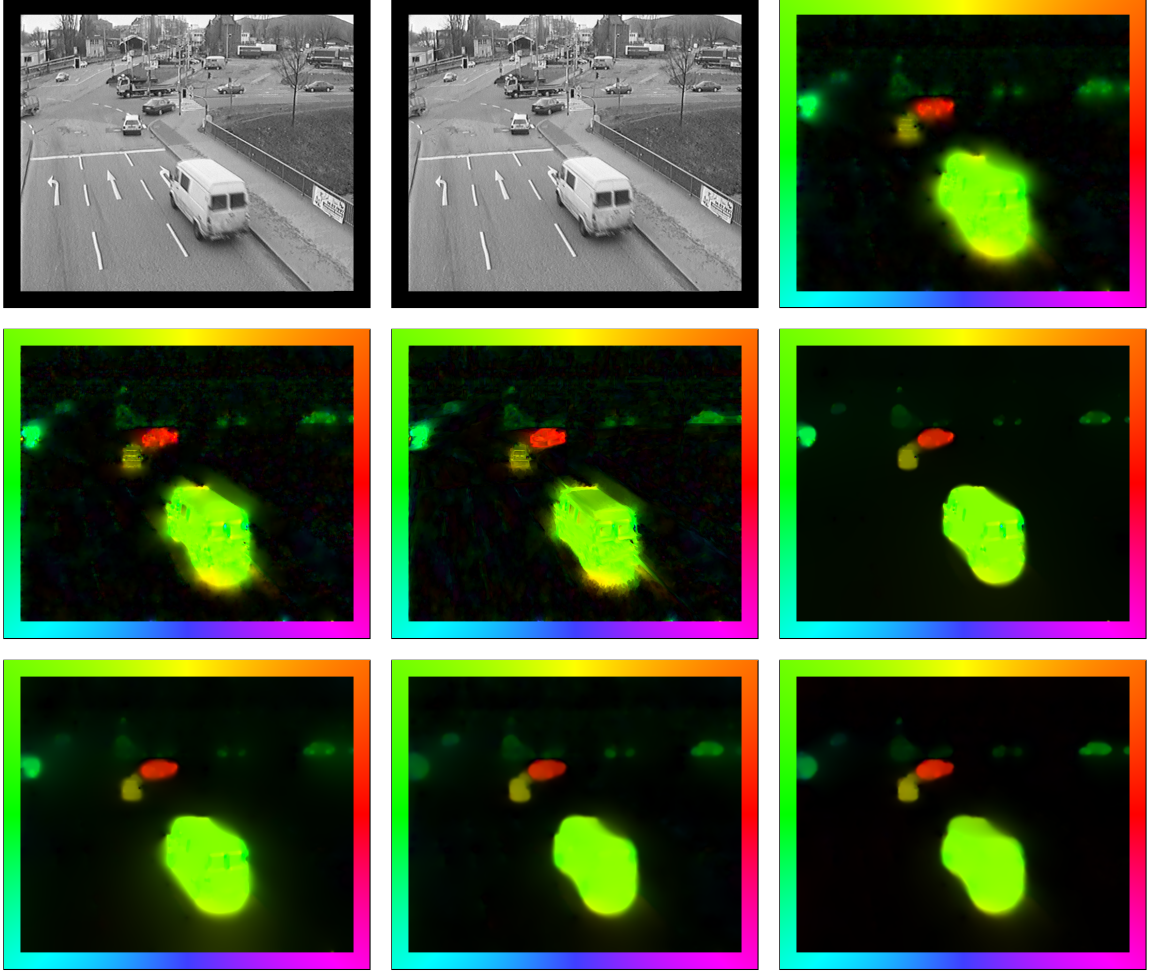
Fig. 3.8: **Experiment II**: Performance of the prototypes for real-world sequences with discontinuities, multiple translational motions and interlacing. **(a) Top Left:** Frame 1130 of the interlaced *Rheinhafen* sequence of size 688×565 (grey-scale). **(b) Top Centre:** Frame 1131. **(c) Top Right:** Prototype A (homogeneous regularisation). **(d) Middle Left:** Prototype B (image-driven isotropic regularisation). **(e) Middle Centre:** Prototype C (image-driven anisotropic regularisation). **(f) Middle Right:** Prototype D (flow-driven isotropic regularisation). **(g) Bottom Left:** Prototype E (flow-driven anisotropic regularisation). **(h) Bottom Centre:** Prototype F (Bruhn *et al.* 2-D, SD). **(i) Bottom Right:** Prototype G (Papenberg *et al.* 3-D, SD).

from a common problem in this sequence: the underestimation of the displacements for the fast driving white van in the foreground. Although the motion field looks reasonable, the actual motion of the white van is much faster. As we have discussed in the context of the Yosemite sequence with clouds, this problem is related to the use of linearised constancy assumptions in the data term.

3.4.2.3 Experiment III: Discontinuities, Multiple Motions, Occlusions, Interlacing

In our third experiment we evaluate the performance of the different prototypes with respect to occlusions. To this end, we consider another popular traffic scene: the *Ettlinger Tor* sequence by Kollnig. As the *Rheinhafen* sequence from the previous experiment, this
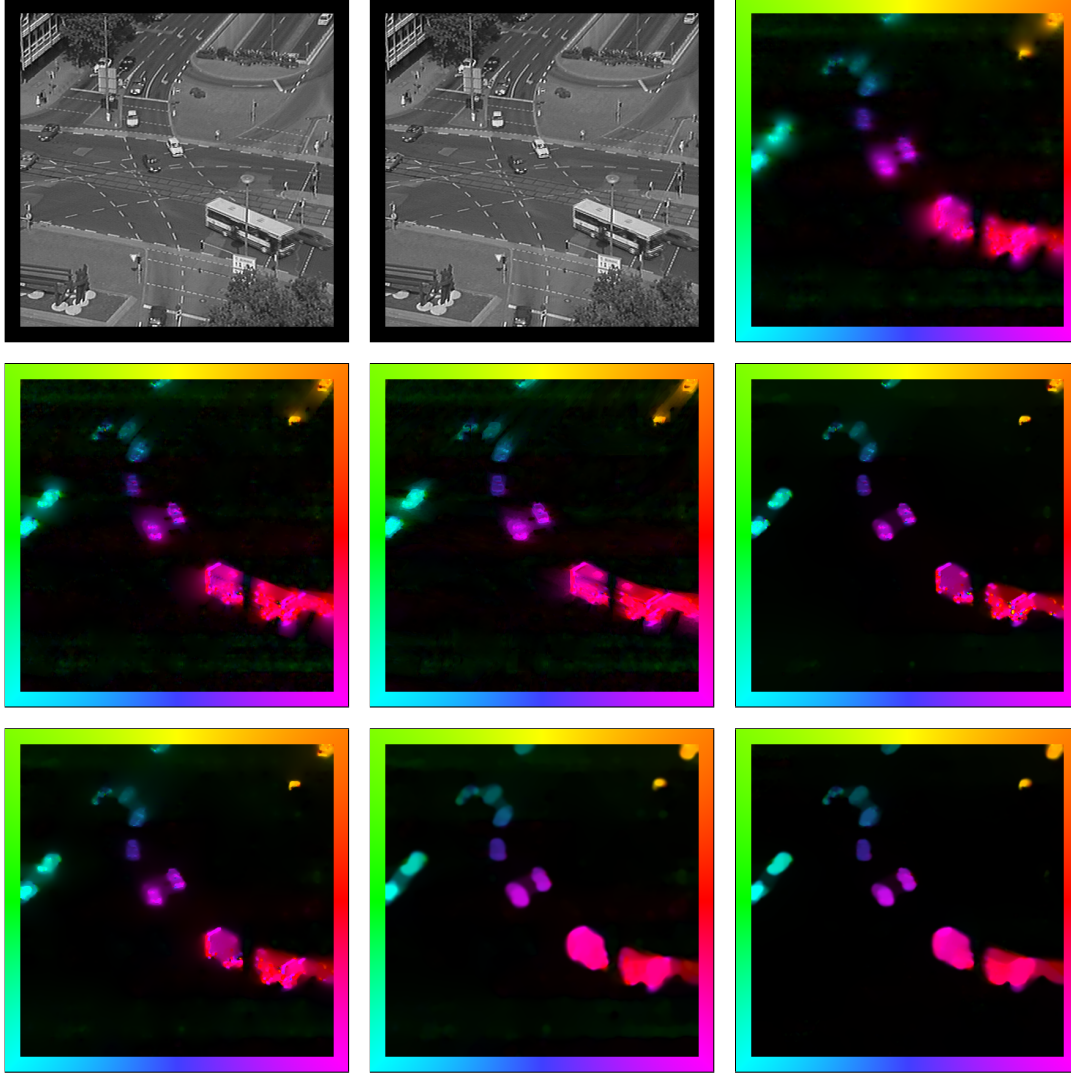
Fig. 3.9: **Experiment III**: Performance of the prototypes for real-world sequences with discontinuities, multiple translational motions, occlusions, and interlacing. **(a) Top Left:** Frame 24 of the *Ettlinger Tor* sequence of size 512×512 (grey-scale). **(b) Top Centre:** Frame 25. **(c) Top Right:** Prototype A (homogeneous regularisation). **(d) Middle Left:** Prototype B (image-driven isotropic regularisation). **(e) Middle Centre:** Prototype C (image-driven anisotropic regularisation). **(f) Middle Right:** Prototype D (flow-driven isotropic regularisation). **(g) Bottom Left:** Prototype E (flow-driven anisotropic regularisation). **(h) Bottom Centre:** Prototype F (Bruhn *et al.* 2-D, SD). **(i) Bottom Right:** Prototype G (Papenberg *et al.* 3-D, SD).

scene is available from `http://i21www.ira.uka.de/image_sequences/` and shows a heavily frequented intersection. However, since it was recorded from a larger distance, objects and displacements are smaller. The sequence consists of 50 frames of size 512×512 and depicts a variety of moving cars (up to 6 pixels per frame). Thereby two cars are of particular interest: a bus in the foreground that passes behind a lamp post and and a car in the upper right area that leaves a tunnel. In both cases occlusions due to static background objects occur and motion discontinuities have to be estimated.

As one can see from the corresponding flow fields in Figure 3.9, all prototypes yield real-

Tab. 3.9: **Comparison To The Literature I:** Comparison between results from the literature with 100 % density and our prototypes. All data refer to the *Yosemite* sequence with cloudy sky. SD = small displacements. HC = constancy assumptions on higher derivatives. NQ-D = non-quadratic (robust) data term. NQ-S = non-quadratic smoothness term or similar strategy (preserves discontinuities). M = multiscale strategy, warping (can handle large displacements). S = segmentation strategy. AAE = average angular error.

Yosemite Sequence with Clouds

technique		HC	NQ-D	NQ-S	3-D	M	S	AAE
Horn/Schunck, orig.	[BFB94]	-	-	-	-	-	-	31.69°
Singh, step 1	[BFB94]	-	-	-	-	-	-	15.28°
Anandan	[BFB94]	-	-	-	-	-	-	13.36°
Singh, step 2	[BFB94]	-	-	-	-	-	-	10.44°
Nagel	[BFB94]	-	-	✓	-	-	-	10.22°
Horn/Schunck, mod.	[BFB94]	-	-	-	-	-	-	9.78°
Uras *et al.*	[BFB94]	-	-	-	-	-	-	8.94°
Prototype A		-	-	-	-	-	-	**7.17°**
Liu *et al.*	[LCR98]	-	-	✓	✓	-	-	6.85 °
Prototype B		-	-	✓	-	-	-	**6.44°**
Prototype E		-	-	✓	-	-	-	**6.42°**
Prototype D		-	-	✓	-	-	-	**6.32°**
Prototype C		-	-	✓	-	-	-	**6.28°**
Prototype F (2-D, SD)		-	✓	✓	-	-	-	**5.74°**
Alvarez *et al.*	[AWS00]	-	-	✓	-	✓	-	5.53°
Mémin/Pérez	[MP98a]	-	✓	✓	-	✓	-	5.38°
Prototype F (3-D, SD)		-	✓	✓	✓	-	-	**5.18°**
Farnebäck	[Far01]	✓	-	-	✓	-	✓	4.84°
Mémin/Pérez	[MP98b]	-	✓	✓	-	✓	-	4.69°
Wu *et al.*	[WaJCL98]	-	-	-	✓	-	-	3.54°
Prototype G (2-D, SD)		✓	✓	✓	-	-	-	**3.50°**
Prototype G (3-D, SD)		✓	✓	✓	✓	-	-	**2.78°**
Teng *et al.*	[TLCH05]	-	✓	✓	✓	✓	-	2.70°
Amiaz/Kiryati	[AK05a]	✓	✓	✓	-	✓	✓	2.04°
Amiaz/Kiryati	[AK05b]	✓	✓	✓	-	✓	✓	1.73°

istic results. However, as in the previous experiment, slight interlacing artifacts are visible in the background for the image-driven approaches A-C. In the case of our flow-driven prototypes D and E, these artifacts are suppressed. However, the flow fields for the different vehicles are not very homogeneous. The best results are once more obtained by our advanced prototypes F and G. Thereby the estimates of the spatiotemporal method G are slightly sharper (see e.g. the two cars at the left border). As before, this is a consequence of the reduced smoothness requirements due to the additional temporal information.

3.4.3   Comparison To the Literature

After we have evaluated the estimation quality of our seven prototypes for a variety of different test scenarios, let us conclude this experimental section with a comparison of

Tab. 3.10: **Comparison To The Literature II:** Comparison between results from the literature with 100 % density and our prototypes. All data refer to the *Yosemite* sequence without cloudy sky. SD = small displacements. HC = constancy assumptions on higher derivatives. NQ-D = non-quadratic (robust) data term. NQ-S = non-quadratic smoothness term or similar strategy (preserves discontinuities). M = multiscale strategy, warping (can handle large displacements). S = segmentation strategy. AAE = average angular error.

Yosemite Sequence without Clouds

technique		HC	NQ-D	NQ-S	3-D	M	S	AAE
Black/Anandan	[BA96]	-	✓	✓	-	✓	-	4.56°
Black	[Bla94]	-	✓	✓	✓	✓	-	3.52°
Szeliski/Coughlan	[SC94]	-	-	-	✓	✓	-	2.45°
Prototype E		-	-	✓	-	-	-	**2.65°**
Prototype A		-	-	-	-	-	-	**2.64°**
Prototype B		-	-	✓	-	-	-	**2.64°**
Prototype D		-	-	✓	-	-	-	**2.58°**
Prototype C		-	-	✓	-	-	-	**2.57°**
Prototype G (2-D, SD)		✓	✓	✓	-	-	-	**2.30°**
Black/Jepson	[BJ96]	-	✓	✓	-	✓	-	2.29°
Prototype F (2-D, SD)		-	✓	✓	-	-	-	**2.27°**
Middendorf	[Mid03]	-	-	✓	✓	-	-	2.27°
Ju *et al.*	[JBJ96]	-	✓	✓	-	✓	✓	2.16°
Bab Hadiashar/Suter	[BHS98]	-	-	-	-	-	-	2.05°
Lai/Vemuri	[LV98]	-	-	-	-	-	-	1.99°
Mémin/Pérez	[MP02]	-	✓	✓	-	✓	✓	1.58°
Teng *et al.*	[TLCH05]	-	✓	✓	✓	✓	-	1.52°
Roth/Black	[RB05]	-	✓	✓	✓	✓	-	1.47°
Prototype F (3-D, SD)		-	✓	✓	✓	-	-	**1.46°**
Prototype G (3-D, SD)		✓	✓	✓	✓	-	-	**1.45°**
Farnebäck	[Far00]	✓	-	-	✓	-	✓	1.40°
Liu *et al.*	[LCR98]	-	-	✓	✓	-	-	1.39 °
Farnebäck	[Far01]	✓	-	-	✓	-	✓	1.14°

the obtained results to the most accurate results from the literature. This is done in the Tables 3.9 and 3.10, where the results for the *Yosemite* sequence *with* and *without* clouds are listed. Thereby all considered approaches have been analysed regarding the use of the following six key concepts: (i) constancy assumptions on higher order derivatives, (ii) robust formulations of the data term, (iii) preservation of discontinuities, (iv) spatiotemporal information, (v) multiscale focusing strategies (warping) and (v) segmentation techniques (e.g. level sets). Moreover, to show the actual performance of our advanced prototypes F and G, results are presented for both spatial (2-D) and spatiotemporal variants (3-D).

3.4.3.1   Comparison I: Yosemite with Clouds

Let us start our comparison with the *Yosemite* sequence *with* clouds (Table 3.9). As one can see, the results for our basic prototype A-E are only average. However, this is not surprising, since these techniques hardly use any of the above mentioned concepts. In

fact, they are comparable to the classical optic flow methods presented in [BFB94]. With respect to the results presented for these methods, our basic approaches perform well. For the advanced prototypes F and G, the situation is completely different. In particular, the spatiotemporal variant of our prototype G gives excellent results. With an average angular error of 2.78° this methods is among the most accurate techniques for estimating the optic flow. One should note that all other state-of-the art methods listed in Table 3.9 are explicitly designed to handle large displacements. Thus, there is still room for improving our prototypes in Chapter 5, where the extension of modelling and numerics to optic flow methods to large displacements is discussed.

3.4.3.2   Comparison II: Yosemite without Clouds

In our second comparison to the literature for the *Yosemite* sequence without clouds tendencies are similar: While the basic prototypes A-E give good results compared to classical approaches such as the ones proposed by Black or Black and Anandan, the advanced prototypes G and F are once again among the leading techniques for estimating the optic flow. One should note that the prototype F now benefits from the fact that no sky and thus no illumination changes are present. As for the sequence with clouds, we can observe that the performance of the spatiotemporal variants of the advanced prototypes is significantly better than the one of the spatial versions. However, also in this case, there is room for improvement by extending our prototypes in such a way that they can handle large displacements.

3.5   Summary

In this chapter we presented seven different prototypes of variational methods for small displacements: five basic variational methods that were chosen to represent the five main regularisation strategies (homogeneous, image-driven isotropic and anisotropic as well as flow-driven isotropic and anisotropic) and two advanced variational techniques that were designed to address two typical problems in motion estimation: noise and varying illumination. While the five basic techniques were directly adapted from the literature, the two advanced methods – the method of Bruhn *et al.* and the method of Papenberg *et al.* – have been developed from scratch. Thereby we demonstrated how the toolbox for constructing variational approaches from the last chapter can actually be applied for the design of novel methods.

As a next step we addressed the minimisation of the presented energy functionals. To this end, we proposed an elliptic strategy and derived the corresponding Euler-Lagrange equations for all seven prototypes. As proposed in the next chapter this was done on the basis of the motion and diffusion tensor notation. This compact representation allowed us to establish a taxonomy of Euler-Lagrange equations for variational optic flow methods. Thereby we distinguished three different types of equations: linear equations, partially nonlinear equations and fully nonlinear equations. For each of the three cases we proposed a suitable discretisation and derived the discrete Euler-Lagrange equations. Moreover, we analysed the resulting linear and nonlinear systems of equation regarding their structure. Thereby we saw that these systems are very sparse and only coupled via a small local neighbourhood. In the next chapter on the numerical solution of these equation system, this property will be exploited to design efficient numerical schemes.

Finally, we evaluated the estimation quality of the selected prototypes for a variety of motion scenarios. We saw that a combination of several concepts within a single energy functional can be very useful: In almost all experiments our advanced methods yielded the best results of all proposed prototypes. In this context, also the general superiority of spatiotemporal regularisers became obvious. All in all, the experiments showed that the appropriate design of variational methods allows to handle a large amount of typical problems in motion estimation. This was also reflected in our final comparison to the literature: The advanced prototypes that we had designed not only gave excellent results, they even competed successfully with state-of-the-art methods that were explicitly constructed for large displacements.

4   Optic Flow For Small Displacements – Solvers

"Not the big will eat the small but the fast will eat the slow."
- Heinz Peter Halek

As we have seen in the previous chapter the minimisation of an energy functional via its Euler-Lagrange equations comes down to solving a large and sparse linear or nonlinear system of equations. To this end, the numerical literature provides us with a large number of iterative methods. The first part of this chapter – Section 4.1 – is dedicated to the solution of *linear* systems of equations. Thereby we present a broad class of methods for which efficient implementations are derived: The Gauß–Seidel and the Successive Overrelaxation method (SOR) representing the class of non-hierarchical iterative solvers [OR00, You71, Mei99], unidirectional coarse-to-fine techniques that stand for the class of simple hierarchical iterative schemes [BD96] and bidirectional multigrid methods that can be identified with the most advanced hierarchical strategies [Bra77, BHM00, Hac85, TOS01, Wes92]. In the second part of the chapter given by Section 4.2 we discuss how these techniques can be extended in such a way, that they can also be used for the solution of *nonlinear* systems of equations. There the focus lies on inexact Lagged-Diffusivity variants of the previous techniques [BBPW04, CM99, Vog02]. Finally, we conclude this chapter with an experimental evaluation of the efficiency of the different numerical schemes that we have proposed before. This is done in Section 4.4.

4.1   The Linear Case

In the following we explain the different numerical schemes for the linear case in detail. This is done in two ways: Firstly, we present the general idea for each scheme based on a rather abstract notation. This notation is very common in the numerical literature, but not related to a specific problem. Secondly, we transfer this knowledge to the field of variational optic flow methods: We show by a concrete example – our prototype A – how these schemes can be applied for our purposes, namely the solution of the previously discussed Euler–Lagrange equations.

General Problem. Let us start by giving a general definition of our problem. In the following we are interested in solving a linear equation system given by

$$A^{\mathbf{h}} \mathbf{x}^{\mathbf{h}} = \mathbf{b}^{\mathbf{h}}, \tag{4.1}$$

where $A^{\mathbf{h}}$ is a symmetric positive definite $2N^{\mathbf{h}} \times 2N^{\mathbf{h}}$ matrix with $O(N^{\mathbf{h}})$ non-zero entries, $\mathbf{x}^{\mathbf{h}}$ is the solution and $\mathbf{b}^{\mathbf{h}}$ is the right hand side. As we have discussed in 3.3.3.1 such linear systems are obtained for our prototypes A-C. The convexity of the underlying energy functionals thereby guarantees a unique solution which is reflected in the invertibility of $A^{\mathbf{h}}$ due to its positive definiteness.

Concrete Example. In our concrete example we are interested in solving the discrete Euler–Lagrange equations that are associated with our prototype A. To this end, let us first identify the different parts of these equations in (3.41) with those of our general formulation in (4.1). This can be done as follows:

$$
\underbrace{\left(\begin{pmatrix} J^{\mathbf{h}}_{11} & J^{\mathbf{h}}_{12} \\ J^{\mathbf{h}}_{12} & J^{\mathbf{h}}_{22} \end{pmatrix} - \alpha \begin{pmatrix} L^{\mathbf{h}}_{\mathrm{L}} & 0 \\ 0 & L^{\mathbf{h}}_{\mathrm{L}} \end{pmatrix} \right)}_{A^{\mathbf{h}}} \underbrace{\begin{pmatrix} \mathbf{u}^{\mathbf{h}}_1 \\ \mathbf{u}^{\mathbf{h}}_2 \end{pmatrix}}_{\mathbf{x}^{\mathbf{h}}} = \underbrace{\begin{pmatrix} -\mathbf{j}^{\mathbf{h}}_{13} \\ -\mathbf{j}^{\mathbf{h}}_{23} \end{pmatrix}}_{\mathbf{b}^{\mathbf{h}}}.
\tag{4.2}
$$

Unfortunately, this formulation of the whole system of equations in block form does not allow to explicitly write down the effect of the discrete linear differential operator $L^{\mathbf{h}}_{\mathrm{L}}$ on $\mathbf{u}^{\mathbf{h}}_1$ and $\mathbf{u}^{\mathbf{h}}_2$. For deriving the iteration instructions for the different numerical schemes, however, this is necessary. Therefore, we switch to a point-based notation, where just the equations for the pixel (i, j) are presented. Knowing that the diffusion tensor for our prototype A is given by the identity matrix, i.e. $\left[T^{\mathbf{h}} \right]_{i,j} = I$, and following the discretisation scheme in (3.24)-(3.27), we finally obtain

$$
\left[J^{\mathbf{h}}_{11} \right]_{i,j} \left[u^{\mathbf{h}}_1 \right]_{i,j} + \left[J^{\mathbf{h}}_{12} \right]_{i,j} \left[u^{\mathbf{h}}_2 \right]_{i,j} - \alpha \sum_{l \in \{x_1, x_2\}} \sum_{\tilde{i}, \tilde{j} \in \mathcal{N}_l(i,j)} \frac{\left(\left[u^{\mathbf{h}}_1 \right]_{\tilde{i}, \tilde{j}} - \left[u^{\mathbf{h}}_1 \right]_{i,j} \right)}{h^2_l} = - \left[J^{\mathbf{h}}_{13} \right]_{i,j} \tag{4.3}
$$

$$
\left[J^{\mathbf{h}}_{12} \right]_{i,j} \left[u^{\mathbf{h}}_1 \right]_{i,j} + \left[J^{\mathbf{h}}_{22} \right]_{i,j} \left[u^{\mathbf{h}}_2 \right]_{i,j} - \alpha \sum_{l \in \{x_1, x_2\}} \sum_{\tilde{i}, \tilde{j} \in \mathcal{N}_l(i,j)} \frac{\left(\left[u^{\mathbf{h}}_2 \right]_{\tilde{i}, \tilde{j}} - \left[u^{\mathbf{h}}_2 \right]_{i,j} \right)}{h^2_l} = - \left[J^{\mathbf{h}}_{23} \right]_{i,j} \tag{4.4}
$$

for $i = 1, ..., N^{\mathbf{h}}_{x_1}$ and $j = 1, ..., N^{\mathbf{h}}_{x_2}$. Here, $\mathcal{N}_l(i, j)$ denotes the set of neighbours of the pixel i, j in direction of axis l. Let us now discuss how this system of equations can be solved.

4.1.1 Splitting-Based Iterative Methods

Although at first glance it might seem desirable to solve the linear system in (4.1) directly by inverting the system matrix, such a proceeding would be much too expensive in terms of both memory requirements and computational costs. However, one can try to approximate this direct approach iteratively by splitting the system matrix in two parts,

$$
A^{\mathbf{h}} = A^{\mathbf{h}}_1 + A^{\mathbf{h}}_2 ,
\tag{4.5}
$$

and introducing an fixed point iteration of type

$$
\mathbf{x}^{\mathbf{h},k+1} = (A^{\mathbf{h}}_1)^{-1} (\mathbf{b}^{\mathbf{h}} - A^{\mathbf{h}}_2 \mathbf{x}^{\mathbf{h},k})
\tag{4.6}
$$

where k denotes the iteration number. Hereby, $(A^{\mathbf{h}}_1)^{-1}$ should be chosen in such a way that it is not only cheap to compute but also a sufficiently good approximation to the true inverse $(A^{\mathbf{h}})^{-1}$. Then, the overall costs to solve the system are not too expensive. This is the core idea of so-called *splitting-based* iterative techniques such as the *Gauß–Seidel* and the *Successive Overrelaxation* method (SOR).

4.1.1.1 The Standard Gauß-Seidel Method

Basic Idea. Despite its poor complexity[1] of $O(n^2)$, the Gauß-Seidel method is one of the most frequently used iterative solvers in the literature [You71, Mei99]. It is easy to implement and does not require a too complex adaptation to a specific problem. Its underlying idea is the decomposition of the original matrix into

$$A^{\mathbf{h}} = \underbrace{(D^{\mathbf{h}} - L^{\mathbf{h}})}_{A_1^{\mathbf{h}}} + \underbrace{(-U^{\mathbf{h}})}_{A_2^{\mathbf{h}}}, \tag{4.7}$$

where $D^{\mathbf{h}}$, $L^{\mathbf{h}}$ and $U^{\mathbf{h}}$ are the diagonal, strictly lower triangular and strictly upper triangular parts of $A^{\mathbf{h}}$. This allows to rewrite the equation system $A^{\mathbf{h}}\mathbf{x}^{\mathbf{h}} = \mathbf{b}^{\mathbf{h}}$ as

$$(D^{\mathbf{h}} - L^{\mathbf{h}})\mathbf{x}^{\mathbf{h}} = (\mathbf{b}^{\mathbf{h}} + U^{\mathbf{h}}\,\mathbf{x}^{\mathbf{h}}) \tag{4.8}$$

and to introduce the fixed point iteration step

$$\mathbf{x}^{\mathbf{h},k+1} = (D^{\mathbf{h}} - L^{\mathbf{h}})^{-1}\,(\mathbf{b}^{\mathbf{h}} + U^{\mathbf{h}}\,\mathbf{x}^{\mathbf{h},k}). \tag{4.9}$$

Due to the lower diagonal structure of the matrix $(D^{\mathbf{h}}-L^{\mathbf{h}})$, the solution for each iteration step can be computed via a simple *forward substitution step*. For problems where $A^{\mathbf{h}}$ has $O(N^{\mathbf{h}})$ entries, this step has linear complexity, i.e. $O(n)$. If we denote by $x_i^{\mathbf{h}}$ the i-th component of a vector $\mathbf{x}^{\mathbf{h}}$, we can formulate the corresponding iteration instructions. In point-notation they are given by

$$x_i^{\mathbf{h},k+1} = (A_{i,j}^{\mathbf{h}})^{-1}\left(\mathbf{b}_i^{\mathbf{h}} - \sum_{j=1}^{i-1} A_{i,j}^{\mathbf{h}}\,x_j^{\mathbf{h},k+1} - \sum_{j=i+1}^{2N^{\mathbf{h}}} A_{i,j}^{\mathbf{h}}\,x_j^{\mathbf{h},k}\right) \tag{4.10}$$

for $i = 1, ..., 2N^{\mathbf{h}}$. By showing that all variables are updated consecutively, this notation makes the dependency of the method on the order of traversal explicit. However, this aspect is less critical, since in most cases, the "wrong" update order does only lead to slight deterioration of convergence rate. One should note that if $A^{\mathbf{h}}$ is positive definite – this is the case for our prototypes A-C – the Gauß-Seidel method converges for all starting vectors $\mathbf{x}^{\mathbf{h},0}$ and for all right hand sides $\mathbf{b}^{\mathbf{h}}$.

Concrete Example. Let us now take a look how the corresponding iteration instructions read for our concrete example. They are given by

---

1. This asymptotical complexity refers to the required number of mathematical operations to compute the solution for simple model problems such as the Poisson equation [TOS01] with n unknowns up to the accuracy of the discretisation.

$$\left[u_1^{\mathbf{h}}\right]_{i,j}^{k+1} = \frac{\displaystyle\sum_{l\in\{x_1,x_2\}} \frac{\alpha}{h_l^2}\left(\sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l^-(i,j)} \left[u_1^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k+1} + \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l^+(i,j)} \left[u_1^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k}\right) - \left[J_{12}^{\mathbf{h}}\right]_{i,j}\left[u_2^{\mathbf{h}}\right]_{i,j}^{k} - \left[J_{13}^{\mathbf{h}}\right]_{i,j}}{\displaystyle\sum_{l\in\{x_1,x_2\}} \frac{\alpha}{h_l^2}|\mathcal{N}_l(i,j)| + \left[J_{11}^{\mathbf{h}}\right]_{i,j}} \quad (4.11)$$

$$\left[u_2^{\mathbf{h}}\right]_{i,j}^{k+1} = \frac{\displaystyle\sum_{l\in\{x_1,x_2\}} \frac{\alpha}{h_l^2}\left(\sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l^-(i,j)} \left[u_2^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k+1} + \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l^+(i,j)} \left[u_2^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k}\right) - \left[J_{12}^{\mathbf{h}}\right]_{i,j}\left[u_1^{\mathbf{h}}\right]_{i,j}^{k+1} - \left[J_{23}^{\mathbf{h}}\right]_{i,j}}{\displaystyle\sum_{l\in\{x_1,x_2\}} \frac{\alpha}{h_l^2}|\mathcal{N}_l(i,j)| + \left[J_{22}^{\mathbf{h}}\right]_{i,j}} \quad (4.12)$$

for $i = 1, ..., N_{x_1}^{\mathbf{h}}$ and $j = 1, ..., N_{x_2}^{\mathbf{h}}$, where $\mathcal{N}_l^-(i,j)$ and $\mathcal{N}_l^+(i,j)$ are the sets of neighbours of pixel (i,j) in direction of axis l that have already been processed and that have still to be processed, respectively. Moreover, $|\mathcal{N}_l(i,j)|$ denotes the number of neighbours of the pixel (i,j) in direction of axis l.

In this context one should note that since we have a linear system of equations, the denominator is independent from $\left[u_1^{\mathbf{h}}\right]_{i,j}$ and $\left[u_2^{\mathbf{h}}\right]_{i,j}$. Consequently, its inverse can be computed a priori so that divisions are not required apart from the set-up-phase (divisions are much more expensive than multiplication, additions and subtractions).

4.1.1.2   Block Gauß-Seidel Variants

Basic Idea. In particular with respect to a strong coupling of unknowns in the equation system, it often makes sense to update more than one variable in the forward substitution step simultaneously. In the literature such schemes are known as *block relaxation methods* [Wes92]. The most popular block relaxation strategies are the *coupled point relaxation* (CPR) and the *alternating line relaxation* (ALR). While the CPR computes all unknowns simultaneously that belong to the same pixel, the ALR determines all those unknowns at the same time that belong to the same line. Since in general the direction of the strongest coupling is neither known nor constant for all unknowns, the ALR method additionally alters the direction when performing several line sweeps. As one can see from their names the CPR and the ALR method implicitly assume that the equation system was obtained from a multi-dimensional problem with several unknowns per pixel. Evidently, this matches our purposes: In the case of optic flow we are looking for a 2-D or 3-D vector field.

Let us assume that we have m blocks and let these blocks of unknowns be given by the sets $\mathcal{B}_i$ for $i = 1, ..., m$. Let furthermore $\mathbf{x}_{\mathcal{B}_i}^{\mathbf{h},k+1}$ be the vector that consists only of those entries of $\mathbf{x}^{\mathbf{h},k+1}$ that coincide with the position of the unknowns from the block $\mathcal{B}_i$. Let finally $A_{\mathcal{B}_i,\mathcal{B}_j}^{\mathbf{h}}$ be the matrix that only contains those entries of $A^{\mathbf{h}}$ that relate the two blocks $\mathcal{B}_i$ and $\mathcal{B}_j$. Then, the associated iteration step for the i-th block $\mathcal{B}_i$ can be formulated as

$$\mathbf{x}_{\mathcal{B}_i}^{\mathbf{h},k+1} = (A_{\mathcal{B}_i,\mathcal{B}_i}^{\mathbf{h}})^{-1}\left(\mathbf{b}_{\mathcal{B}_i}^{\mathbf{h}} - \sum_{j=1}^{i-1} A_{\mathcal{B}_i,\mathcal{B}_j}^{\mathbf{h}}\,\mathbf{x}_{\mathcal{B}_i}^{\mathbf{h},k+1} - \sum_{j=i+1}^{m} A_{\mathcal{B}_i,\mathcal{B}_j}^{\mathbf{h}}\,\mathbf{x}_{\mathcal{B}_i}^{\mathbf{h},k}\right). \quad (4.13)$$

One should note that if each block $\mathcal{B}_i$ only consists of one unknown, namely $x_i^{\mathbf{h}}$, this scheme comes down to the standard Gauß-Seidel method (4.10).

Block Gauß-Seidel Variants - Concrete Example I. Let us now take a look at the iterations instructions in point-form for the Gauß-Seidel method with coupled point relaxation (CPR). They are given by

$$
\begin{pmatrix} \left[u_1^{\mathbf{h}}\right]_{i,j}^{k+1} \\ \left[u_2^{\mathbf{h}}\right]_{i,j}^{k+1} \end{pmatrix} = \begin{pmatrix} \left[K_{11}^{\mathbf{h}}\right]_{i,j} & \left[K_{12}^{\mathbf{h}}\right]_{i,j} \\ \left[K_{12}^{\mathbf{h}}\right]_{i,j} & \left[K_{22}^{\mathbf{h}}\right]_{i,j} \end{pmatrix}^{-1} \begin{pmatrix} \left[b_1^{\mathbf{h}}\right]_{i,j}^{k} \\ \left[b_2^{\mathbf{h}}\right]_{i,j}^{k} \end{pmatrix} \tag{4.14}
$$

for $i = 1, ..., N_{x_1}^{\mathbf{h}}$ and $j = 1, ..., N_{x_2}^{\mathbf{h}}$ with the matrix entries

$$
\left[K_{11}^{\mathbf{h}}\right]_{i,j} = \left[J_{11}^{\mathbf{h}}\right]_{i,j} + \sum_{l \in \{x_1, x_2\}} \frac{\alpha}{h_l^2} |\mathcal{N}_l(i,j)| \tag{4.15}
$$

$$
\left[K_{12}^{\mathbf{h}}\right]_{i,j} = \left[J_{12}^{\mathbf{h}}\right]_{i,j} \tag{4.16}
$$

$$
\left[K_{22}^{\mathbf{h}}\right]_{i,j} = \left[J_{22}^{\mathbf{h}}\right]_{i,j} + \sum_{l \in \{x_1, x_2\}} \frac{\alpha}{h_l^2} |\mathcal{N}_l(i,j)| \tag{4.17}
$$

and right hand side

$$
\left[b_1^{\mathbf{h}}\right]_{i,j}^{k} = -\left[J_{13}^{\mathbf{h}}\right]_{i,j} + \sum_{l \in \{x_1, x_2\}} \frac{\alpha}{h_l^2} \left(\sum_{\tilde{i},\tilde{j} \in \mathcal{N}_l^-(i,j)} \left[u_1^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k+1} + \sum_{\tilde{i},\tilde{j} \in \mathcal{N}_l^+(i,j)} \left[u_1^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k} \right) \tag{4.18}
$$

$$
\left[b_2^{\mathbf{h}}\right]_{i,j}^{k} = -\left[J_{23}^{\mathbf{h}}\right]_{i,j} + \sum_{l \in \{x_1, x_2\}} \frac{\alpha}{h_l^2} \left(\sum_{\tilde{i},\tilde{j} \in \mathcal{N}_l^-(i,j)} \left[u_2^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k+1} + \sum_{\tilde{i},\tilde{j} \in \mathcal{N}_l^+(i,j)} \left[u_2^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k} \right) . \tag{4.19}
$$

One should note that apart from discretisation details, this relaxation scheme is identical to the one proposed by Horn and Schunck in [HS81]. As for the standard Gauss-Seidel method, also in this case no divisions are required. This is due to the fact that the entries of the 2×2 matrices

$$
M_{i,j}^{\mathbf{h}} = \begin{pmatrix} \left[K_{11}^{\mathbf{h}}\right]_{i,j} & \left[K_{12}^{\mathbf{h}}\right]_{i,j} \\ \left[K_{12}^{\mathbf{h}}\right]_{i,j} & \left[K_{22}^{\mathbf{h}}\right]_{i,j} \end{pmatrix}^{-1} \tag{4.20}
$$

do not depend on the unknowns $\left[u_1^{\mathbf{h}}\right]_{i,j}$ and $\left[u_2^{\mathbf{h}}\right]_{i,j}$ and thus can be computed in advance. As a consequence, the computational effort of the block is comparable to that of the standard method.

Concrete Example II. As second example let us now investigate the structure of the iteration instructions for the Gauß-Seidel method with x_2-line relaxation. To this end, we need to address whole lines of unknowns. This can be realised by using the following abbreviations (in accordance with (3.38) - (3.40)):

$$
\left[\mathbf{u}_l^{\mathbf{h}}\right]_i^{k} := \left(\left[u_l^{\mathbf{h}}\right]_{i,1}^{k}, \ldots, \left[u_l^{\mathbf{h}}\right]_{i,N_{x_2}^{\mathbf{h}}}^{k} \right)^{\top} \qquad (l = 1, 2), \tag{4.21}
$$

$$
\left[\mathbf{j}_{mn}^{\mathbf{h}}\right]_i := \left(\left[J_{mn}\right]_{i,1}^{\mathbf{h}}, \ldots, \left[J_{mn}\right]_{i,N_{x_2}^{\mathbf{h}}}^{\mathbf{h}} \right)^{\top} \qquad (m, n = 1, 2, 3), \tag{4.22}
$$

$$
\left[J_{mn}^{\mathbf{h}}\right]_i := \mathrm{diag}\left(\left[\mathbf{j}_{mn}^{\mathbf{h}}\right]_i^{\top} \right) \qquad (m, n = 1, 2, 3). \tag{4.23}
$$

Consequently, the iteration instructions are also given in line-form. They read

$$\left[\mathbf{u}_1^{\mathbf{h}}\right]_i^{k+1} \;\; = \;\; \left[K_1^{\mathbf{h}}\right]_i^{-1} \left[\mathbf{b_1}^{\mathbf{h}}\right]_i^{k} \; , \tag{4.24}$$

$$\left[\mathbf{u}_2^{\mathbf{h}}\right]_i^{k+1} \;\; = \;\; \left[K_2^{\mathbf{h}}\right]_i^{-1} \left[\mathbf{b_2}^{\mathbf{h}}\right]_i^{k} \tag{4.25}$$

for $i = 1, ..., N_{x_1}^{\mathbf{h}}$ with the $N_{x_2}^{\mathbf{h}} \times N_{x_2}^{\mathbf{h}}$ matrices

$$\left[K_1^{\mathbf{h}}\right]_i \;\; = \;\; -L_{x_2 x_2} + \left[J_{11}^{\mathbf{h}}\right]_i \; , \tag{4.26}$$

$$\left[K_2^{\mathbf{h}}\right]_i \;\; = \;\; -L_{x_2 x_2} + \left[J_{22}^{\mathbf{h}}\right]_i \; , \tag{4.27}$$

where

$$L_{x_2 x_2} = \frac{\alpha}{h_{x_2}^2} \begin{pmatrix} -1 & 1 & & & & 0 \\ 1 & -2 & 1 & & & \\ & 1 & \cdot & & \cdot & \\ & & & \cdot & -2 & 1 \\ 0 & & & & 1 & -1 \end{pmatrix} , \tag{4.28}$$

and the right hand side vectors

$$\left[\mathbf{b_1}^{\mathbf{h}}\right]_i^{k} \;\; = \;\; \frac{\alpha}{h_{x_1}^2} I \left(\sum_{\tilde{i} \in \mathcal{N}_{x_1}^{-}(i)} \left[\mathbf{u}_1^{\mathbf{h}}\right]_{\tilde{i}}^{k+1} + \sum_{\tilde{i} \in \mathcal{N}_{x_1}^{+}(i)} \left[\mathbf{u}_1^{\mathbf{h}}\right]_{\tilde{i}}^{k} \right) - \left[J_{12}^{\mathbf{h}}\right]_i \left[\mathbf{u}_2^{\mathbf{h}}\right]_i^{k} - \left[\mathbf{j}_{13}^{\mathbf{h}}\right]_i \quad (4.29)$$

$$\left[\mathbf{b_2}^{\mathbf{h}}\right]_i^{k} \;\; = \;\; \frac{\alpha}{h_{x_1}^2} I \left(\sum_{\tilde{i} \in \mathcal{N}_{x_1}^{-}(i)} \left[\mathbf{u}_2^{\mathbf{h}}\right]_{\tilde{i}}^{k+1} + \sum_{\tilde{i} \in \mathcal{N}_{x_1}^{+}(i)} \left[\mathbf{u}_2^{\mathbf{h}}\right]_{\tilde{i}}^{k} \right) - \left[J_{12}^{\mathbf{h}}\right]_i \left[\mathbf{u}_1^{\mathbf{h}}\right]_i^{k+1} - \left[\mathbf{j}_{23}^{\mathbf{h}}\right]_i \; . (4.30)$$

Here, I is the identity matrix and $\mathcal{N}_l^{-}(i)$ and $\mathcal{N}_l^{+}(i)$ are the sets of neighbouring lines in direction of axis l that have already been processed and that have still to be processed, respectively. As one can see, the solution for each line of unknowns requires the solution of two tri-diagonal systems of equations. However, in contrast to the point-coupled case, inverting these systems in advance is not a good idea. Since the matrices $\left[K_1^{\mathbf{h}}\right]_i$ and $\left[K_2^{\mathbf{h}}\right]_i$ are diagonally dominant and have only negative off-diagonal elements, the resulting inverse would be a dense matrix of size $N_{x_2}^{\mathbf{h}} \times N_{x_2}^{\mathbf{h}}$ [Var00;p.94]. Instead, one can apply the Thomas algorithm [Tho49], an efficient variant of the Gaussian algorithm [Sch97] for sparse matrices with complexity $O(n)$. In this case, at least that part of the *Thomas algorithm* can be computed in advance that does not depend on the right hand side (which changes during the iterations). One should note that the use of such (inner) solvers with linear complexity for each line does not increase the overall complexity of the underlying Gauß-Seidel method.

In the case of Gauß-Seidel method with alternating line relaxation (ALR), this x_2-line sweep is followed by a relaxation in x_1-direction, which can be derived in an analogous

way. An interesting aspect at this point is the fact, that in our equation system one direction is not covered by such a standard alternating relaxation scheme: the direction of the different unknowns per pixels ($\left[u_1^{\mathbf{h}}\right]_{i,j}$ and $\left[u_2^{\mathbf{h}}\right]_{i,j}$). Therefore it is useful to complement the two line relaxation steps by a point-coupled one.

4.1.1.3   The Successive Overrelaxation (SOR) Method

Basic Idea. Apart from using block relaxation techniques to improve the performance for strongly coupled systems of equations, one may also think of speeding up the general convergence of the Gauß–Seidel method. This can be done by means of the *Successive Overrelaxation* (SOR) technique which has in the optimal case a complexity[2] of $O(n^{1.5})$. It is based on an extrapolation of the Gauß–Seidel results [You71] and can be written as

$$
\mathbf{x}_i^{\mathbf{h},k+1} \;=\; (1-\omega)\,\mathbf{x}_i^{\mathbf{h},k} + \omega\,\underbrace{(A_{i,j}^{\mathbf{h}})^{-1}\Big(\mathbf{b}_i^{\mathbf{h}} - \sum_{j=1}^{i-1} A_{i,j}^{\mathbf{h}}\, x_j^{\mathbf{h},k+1} - \sum_{j=i+1}^{2N^{\mathbf{h}}} A_{i,j}^{\mathbf{h}}\, x_j^{\mathbf{h},k}\Big)}_{\text{original Gauß-Seidel result}}
$$

$$(4.31)$$

for $i = 1, ..., 2N^{\mathbf{h}}$, where ω is a *relaxation parameter* in the interval $(0, 2)$. With $D^{\mathbf{h}}$, $L^{\mathbf{h}}$ and $U^{\mathbf{h}}$ as defined as before, these iteration instruction can also be reformulated in matrix notation:

$$
\mathbf{x}^{\mathbf{h},k+1} = (D^{\mathbf{h}} - \omega L^{\mathbf{h}})^{-1}\Big(\omega\,\mathbf{b}^{\mathbf{h}} + \big((1-\omega)D^{\mathbf{h}} + \omega U^{\mathbf{h}}\big)\mathbf{x}^{\mathbf{h},k}\Big)\,.
\qquad (4.32)
$$

As one can see, for $\omega = 1$ the SOR technique comes down to the Gauß-Seidel method. However, in practice, other values for ω are more interesting: While schemes based on a *smaller value* actually perform underrelaxation (*dampened Gauß-Seidel method*) and thus may offer improved stability as well as better smoothing properties compared to the original method [Wes92, TOS01], methods based on a *larger value* perform real overrelaxation and thus may accelerate the convergence significantly. Thereby the actual choice of ω is decisive: A well-chosen ω may allow to speed up the computation by *one or two orders of magnitude*. Since, in general, it is not possible to compute the optimal value for ω in advance, it has to be determined either empirically (for a specific model problem) or heuristically (based on some kind of estimate). As the Gauß-Seidel method, the SOR method converges for our prototypes A-C, since the underlying system matrix is symmetric and positive definite.

Concrete Example. After we have discussed the general idea of the SOR method, let us now study how this scheme looks like for our concrete example. Analogously to the iteration instructions for the Gauß-Seidel method, the SOR instructions are given by

---

2. This asymptotical complexity refers to simple model problems such as the Poisson equation [TOS01].

$$\left[u_1^{\mathbf{h}}\right]_{i,j}^{k+1} = (1-\omega)\left[u_1^{\mathbf{h}}\right]_{i,j}^{k} \tag{4.33}$$

$$+\,\omega\,\frac{\displaystyle\sum_{l\in\{x_1,x_2\}}\frac{\alpha}{h_l^2}\left(\displaystyle\sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l^-(i,j)}\left[u_1^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k+1}+\displaystyle\sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l^+(i,j)}\left[u_1^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k}\right)-\left[J_{12}^{\mathbf{h}}\right]_{i,j}\left[u_2^{\mathbf{h}}\right]_{i,j}^{k}-\left[J_{13}^{\mathbf{h}}\right]_{i,j}}{\displaystyle\sum_{l\in\{x_1,x_2\}}\frac{\alpha}{h_l^2}|\mathcal{N}_l(i,j)|+\left[J_{11}^{\mathbf{h}}\right]_{i,j}}\,,$$

$$\left[u_2^{\mathbf{h}}\right]_{i,j}^{k+1} = (1-\omega)\left[u_2^{\mathbf{h}}\right]_{i,j}^{k} \tag{4.34}$$

$$+\,\omega\,\frac{\displaystyle\sum_{l\in\{x_1,x_2\}}\frac{\alpha}{h_l^2}\left(\displaystyle\sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l^-(i,j)}\left[u_2^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k+1}+\displaystyle\sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l^+(i,j)}\left[u_2^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k}\right)-\left[J_{12}^{\mathbf{h}}\right]_{i,j}\left[u_1^{\mathbf{h}}\right]_{i,j}^{k+1}-\left[J_{23}^{\mathbf{h}}\right]_{i,j}}{\displaystyle\sum_{l\in\{x_1,x_2\}}\frac{\alpha}{h_l^2}|\mathcal{N}_l(i,j)|+\left[J_{22}^{\mathbf{h}}\right]_{i,j}}\,.$$

for $i = 1, ..., N_{x_1}^{\mathbf{h}}$ and $j = 1, ..., N_{x_2}^{\mathbf{h}}$. As in the previous cases where the inverse of the diagonal elements/block has not been too large, we can compute the whole denominator in advance to avoid divisions. Thus, from a computational point of view, the cost for using the SOR scheme are only marginally higher than those for using the Gauß-Seidel method. However, one should keep in mind that the SOR methods may need significantly less iterations to converge.

4.1.2   Unidirectional Multigrid Methods

So far, we have only considered numerical methods that work on a *single grid*. However, these methods perform relatively poor if the equation system is only coupled via a small local neighbourhood (such as the discrete differential operator $L_{\mathrm{L}}^{\mathbf{h}}$). In fact, it may take thousands of iterations to transport local information between unknowns that are not coupled directly. A popular remedy to speed up the computation in those cases is the use of *unidirectional multigrid methods* [BA91, BD96].

Basic Idea. Such methods are based on the observation that the required number of iterations may be significantly reduced if the initialisation is already close enough to the correct solution. In order to obtain such a good initialisation, they follow a simple two-grid strategy: Instead of solving the equation system on the original (fine) grid they first transfer this system to a *coarser grid* where it becomes

$$A^{\mathbf{H}}\mathbf{x}^{\mathbf{H}} = \mathbf{b}^{\mathbf{H}}. \tag{4.35}$$

Here, $\mathbf{H} = (H_{x_1}, H_{x_2})^{\top}$ denotes the coarse grid cell size with $H_{x_1} > h_{x_1}$ and $H_{x_2} > h_{x_2}$. After solving this system with one of the proposed non-hierarchical iterative methods, the obtained solution is then interpolated to the fine grid, where it serves as good initial guess for the original problem. This initialisation strategy by means of a coarse grid solution offers two advantages: On one hand, the equation system on the coarse grid can be solved much cheaper than the one on the fine grid. Assuming $H_{x_1} \approx 2h_{x_1}$ and $H_{x_2} \approx 2h_{x_2}$, the additional computational costs are only 25%. On the other hand it is plausible that the

interpolated solution from the coarse grid like constitutes a reasonable initialisation for the fine grid solver. It is therefore not surprising that in general much more than the required 25% of all fine grid iterations are saved to make the approach worthwhile.

Intergrid Transfer Operators. Before we discuss how the required coarse grid equation system can be derived, let us first address another problem that arises if more than one grid is used: The transfer of data and solution between the different grids. To this end, *a pair* of *intergrid transfer operators* is needed:

- *Restriction Operator*. Firstly, one has to define a so-called *restriction operator* $R^{\mathbf{h}\to\mathbf{H}}$ that transfers *data* from the fine to the coarse grid:

$$\mathbf{b}^{\mathbf{H}} = R^{\mathbf{h}\to\mathbf{H}}\mathbf{b}^{\mathbf{h}}. \tag{4.36}$$

  Such an operator usually combines a pure resampling strategy (injection) with an appropriate amount of smoothing (low-pass filtering) to overcome the aliasing problem. A good choice in this context is the *full-weighting* operator [BHM00] or the operator for *area based averaging* [BWF+05].

- *Prolongation Operator*. Apart from a restriction operator also a *prolongation operator* $P^{\mathbf{H}\to\mathbf{h}}$ is required that transfers the *solution* from the coarse to the fine grid:

$$\mathbf{x}^{\mathbf{h}} = P^{\mathbf{H}\to\mathbf{h}}\mathbf{x}^{\mathbf{H}}. \tag{4.37}$$

  In this context, *constant* or *linear* interpolation are most frequently used [Wes92, BHM00]. As shown in [BWF+05] also *area based interpolation* may give good results. If computational time is not important, also more accurate (and more costly) interpolation schemes may be considered such as cubic splines or operator dependent interpolation (combined with operator dependent restriction) [Wes92].

One should note that although that although we have to restrict image data from the fine to the coarse grid, the solution is only transferred in *one direction*, namely from the coarse grid to the fine grid. This property gives the whole class of unidirectional multigrid methods their name.

Coarse Grid Equation System. After we have discussed the intergrid transfer operators, let us now investigate suitable strategies how the actual coarse grid equation system can be derived. To this end, three different approaches are proposed in the literature: the *Problem Based Coarse Grid Approximation* (PCA), the *Discretisation Coarse Grid Approximation* (DCA) and the *Galerkin Coarse Grid Approximation* (GCA) [BHM00].

- *Problem Based Coarse Grid Approximation* (PCA). The first and simplest approach is based on the *rediscretisation of the original problem*. This is usually done by

downsampling any problem relevant data and discretising the continuous Euler-Lagrange equations using the coarse grid cell size $\mathbf{H}$. Although such a proceeding is very transparent – it allows to interpret the obtained coarse grid equation system as a separate problem with different grid size – its approximation quality with respect to the fine grid solution is relatively poor. This is intuitively clear, since the approach does not tackle the actual task directly: the downsampling of the equation system.

- *Discretisation Coarse Grid Approximation* (DCA): The second and often cheapest approach in terms of computational costs is the Discretisation Coarse Grid Approximation. Its basic idea is to *rediscretise the discrete Euler-Lagrange equations* directly. The obtained coarse grid equation systems is in general a much better approximation to the fine grid equation system as in the case of the PCA. This is not surprising, since in particular with respect to coefficients that depend nonlinearly on the input data, the first approach introduces additional errors.

- *Galerkin Coarse Grid Approximation* (GCA): The third and last approach is based on the Galerkin condition [BHM00]. It tackles the problem directly by setting

$$A^{\mathbf{H}} = R^{\mathbf{h}\to\mathbf{H}} A^{\mathbf{h}} P^{\mathbf{H}\to\mathbf{h}}. \tag{4.38}$$

and

$$\mathbf{b}^{\mathbf{H}} = R^{\mathbf{h}\to\mathbf{H}} \mathbf{b}^{\mathbf{h}} \tag{4.39}$$

where $R^{\mathbf{h}\to\mathbf{H}}$ and $P^{\mathbf{H}\to\mathbf{h}}$ are the restriction and prolongation operator as defined before. One should note that applying the coarse matrix $A^{\mathbf{H}}$ to a vector $\mathbf{x}^{\mathbf{H}}$ comes down then to prolonging that vector, applying the original matrix, and restricting it again. Although this is the approach that offers the most direct and thus most accurate approximation strategy, it is also by far the most expensive one. Even using the efficient multiplication scheme for three sparse matrices proposed in [Wes92], the computation of the coarse grid equation systems takes about ten times longer than in the case of the PCA or DCA .

While the first strategy (PCA) is very popular in the computer vision community due to its simplicity and transparency – see e.g. [Ana89, BAK91, LKW94] in the context of optic flow – the other two (better performing) concepts are mainly found in the numerical literature [Wes92, BD96, BHM00]. For our purpose the discretisation coarse grid approximation is the most interesting one: It offers a good approximation quality at low computational costs.

Hierarchical Application. So far we have only considered the two-grid case, where a coarser grid is used to provide the finer one with a suitable initialisation. However, in order to speed up the computation even further, one may think of using a *third*, even coarser grid whose solution serves as initial guess for the second (coarse) one. Such a hierarchical initialisation strategy with more than two grids is also referred to as *cascadic multigrid* [BD96]. In Figure 4.1 this idea is illustrated by a schematic drawing for four levels. As one can see, starting from a coarser version the original problem is refined step by step. Thereby coarser solutions serve as initialisation for finer levels.
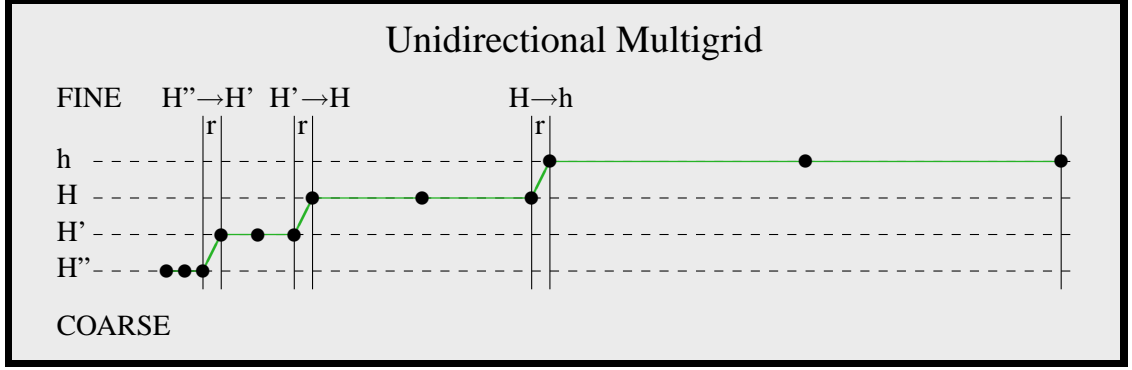
Fig. 4.1: Schematic drawing of a unidirectional multigrid strategy. Starting from a very coarse variant of the original equation system, the solution is refined step by step. The corresponding refinements steps are marked with 'r'. At each resolution level several iterations with the basic solver are performed. This is indicated by large black dots.

Concrete Example. As we have indicated before, the discretisation coarse grid approximation (DCA) offers the best ratio between approximation quality and computational costs. Therefore, let us now demonstrate how a corresponding unidirectional multigrid scheme can be derived for our prototype A. Its construction is done in four steps.

I) *Basic Solver.* Firstly, we have to choose a basic iterative solver. Since our prototype A (homogenous regularisation) does not require the handling of anisotropic neighbourhood coupling, we propose the use of a *Gauß-Seidel method with coupled point relaxation* (CPR). The corresponding iteration instructions are given by (4.14 - 4.19). The same solver is also useful for our prototype B (isotropic regularisation). Only in the case of prototype C (anisotropic regularisation), a Gauß-Seidel method with alternating line relaxation (ALR) may be appropriate.

II) *Coarse Grid Cell Size.* Secondly, we have to define the new coarse grid size **H**. Since our approach shall be applicable to *any* problem size it should not have the typical limitation to sequences with image sizes that are a power of 2. Therefore we propose the following computation of $H_{x_1} \times H_{x_2}$. Let $N_{x_1}^{\mathbf{h}}$ and $N_{x_2}^{\mathbf{h}}$ be the number of cells on the fine grid in x and y direction. Then the new cell size is given by

$$H_{x_1} := h_{x_1} \frac{N_{x_1}^{\mathbf{h}}}{N_{x_1}^{\mathbf{H}}} \qquad\qquad H_{x_2} := h_{x_2} \frac{N_{x_2}^{\mathbf{h}}}{N_{x_2}^{\mathbf{H}}} \qquad\qquad (4.40)$$

with $N_{x_1}^H = \lceil N_{x_1}^{\mathbf{h}}/2 \rceil$ and $N_{x_2}^{\mathbf{H}} = \lceil N_{x_2}^{\mathbf{h}}/2 \rceil$, where $\lceil z \rceil$ is the smallest integer number m with $m \geq z$. Thus the total number of cells at the coarser grid is $N^{\mathbf{H}} = N_{x_1}^{\mathbf{H}} N_{x_2}^{\mathbf{H}}$.

III) *Intergrid Transfer Operators.* Thirdly, we have to define the interpolation and restriction operator. Following the idea of [BWF+05, BWKS06] we chose *area based interpolation* and *area based averaging*. As simple example for these operators is given in Figure 4.2.
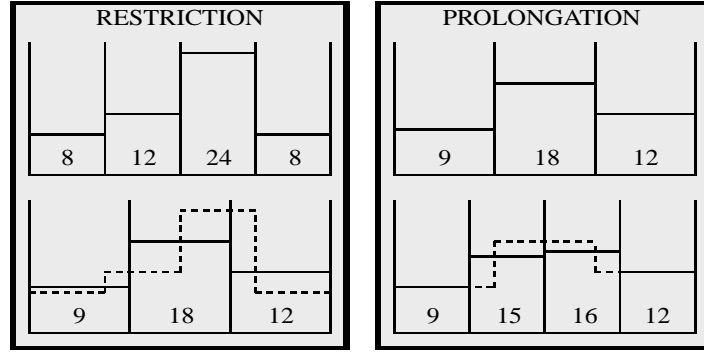
Fig. 4.2: One-dimensional example for the an area based restriction operator $R^{\mathbf{h}\to\mathbf{H}}$ and an area based prolongation operator $P^{\mathbf{H}\to\mathbf{h}}$ with $\frac{\mathbf{h}}{\mathbf{H}} = \frac{3}{4}$.

IV) *Coarse Grid Equation System.* Finally, we have to rediscretise the discrete Euler–Lagrange equations. This yields the coarse grid equation system

$$\left[J_{11}^{\mathbf{H}}\right]_{i,j}\left[u_1^{\mathbf{h}}\right]_{i,j} + \left[J_{12}^{\mathbf{H}}\right]_{i,j}\left[u_2^{\mathbf{H}}\right]_{i,j} - \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)} \frac{\left(\left[u_1^{\mathbf{H}}\right]_{\tilde{i},\tilde{j}} - \left[u_1^{\mathbf{H}}\right]_{i,j}\right)}{H_l^2} = -\left[J_{13}^{\mathbf{H}}\right]_{i,j},$$

(4.41)

$$\left[J_{12}^{\mathbf{h}}\right]_{i,j}\left[u_1^{\mathbf{h}}\right]_{i,j} + \left[J_{22}^{\mathbf{H}}\right]_{i,j}\left[u_1^{\mathbf{H}}\right]_{i,j} - \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)} \frac{\left(\left[u_2^{\mathbf{H}}\right]_{\tilde{i},\tilde{j}} - \left[u_2^{\mathbf{h}}\right]_{i,j}\right)}{H_l^2} = -\left[J_{23}^{\mathbf{H}}\right]_{i,j},$$

(4.42)

for $i = 1, ..., N_{x_1}^{\mathbf{H}}$ and $j = 1, ..., N_{x_2}^{\mathbf{H}}$. Thereby the entries of the coarse grid motion tensor $\left[J^{\mathbf{H}}\right]_{i,j}$ are obtained via channelwise restriction:

$$\left[J_{mn}^{\mathbf{H}}\right]_{i,j} = R^{\mathbf{h}\to\mathbf{H}}\left[J_{mn}^{\mathbf{h}}\right]_{i,j} \qquad (m, n = 1, 2, 3). \tag{4.43}$$

One should note that the choice of the restriction operator ensures that these coarse grid tensors remain positive semidefinite. This in turn guarantees the positive definiteness of the resulting coarse grid equation system so that the same non-hierarchical iterative solvers can be applied as on the fine grid. Moreover, one should note that $\left[J_{mn}^{\mathbf{H}}\right]_{i,j} \neq \left[f_{x_n}^{\mathbf{H}}\right]_{i,j}\left[f_{x_m}^{\mathbf{H}}\right]_{i,j} = \left[\widehat{J}_{nm}^{\mathbf{H}}\right]_{i,j}$. This less accurate variant $\left[\widehat{J}_{nm}^{\mathbf{H}}\right]_{i,j}$ of the coarse grid motion tensor would be obtained by the problem based discretisation approach (PCA). Finally, one should note that for rediscretising the linear differential operator $L_{\mathrm{L}}^{\mathbf{h}}$ via the scheme (3.24)-(3.27), also the diffusion tensors have to be transferred to the coarse grid. Analogously to the motion tensors this can be done by channelwise restriction

$$\left[T_{mn}^{\mathbf{H}}\right]_{i,j} = R^{\mathbf{h}\to\mathbf{H}}\left[T_{mn}^{\mathbf{h}}\right]_{i,j} \qquad (m, n = 1, 2). \tag{4.44}$$

For our prototype A this comes down to restricting the identity matrix which remains the identity matrix.

This example for a unidirectional multigrid scheme shows the advantages of our motion and diffusion tensor notation explicitly: *The construction of efficient hierarchical methods becomes very simple.* As we will see in the following, this does also apply to much more advanced multigrid techniques.

4.1.3 Bidirectional Multigrid Methods

In the previous section, we have discussed *unidirectional* multigrid methods that speed up the computation by using a hierarchical initialisation strategy. However, from a numerical viewpoint, these unidirectional coarse-to-fine schemes are not the end of the road. Very promising – in particular in the context of variational methods – are so-called *bidirectional* multigrid methods [Bra77, BHM00, Hac85, TOS01, Wes92]. These techniques that create a sophisticated hierarchy of equation systems with excellent error reduction properties belong to the fastest numerical schemes for solving linear or nonlinear systems of equations. In contrast to unidirectional schemes they revisit coarser resolution levels in order to obtain useful correction steps. Thus, they are able to overcome the typical limitation of basic iterative solvers that is also present in unidirectional multigrid schemes: *the weak attenuation of low error frequencies* [Bra77, Hac85]. Moreover, they can benefit from all advantages of unidirectional multigrid methods, since they may use the same coarse-to-fine initialisation strategy on top. Then, these bidirectional multigrid schemes are referred to as *full* multigrid methods [BHM00].

Local Mode Analysis. In order to understand the basic idea behind bidirectional multigrid methods we have to ask ourselves the following question: Why do non-hierarchical iterative solvers such as the standard Gauß–Seidel method usually perform so poorly when solving equation systems that are coupled only via a small local neighbourhood? A suitable tool to answer this question is the so-called *local mode analysis* [BHM00]. This analysis investigates the development of the different error frequencies in the Fourier domain. Hereby, the term "error" refers to the difference between the *approximative* numerical solution for the equation system (obtained after a certain number of iterations) and its *correct* numerical solution. If a method converges, this error decreases during the iterations until the level of quantisation is reached. In the Fourier domain of the error such a behaviour is reflected in an attenuation of the different frequencies. How strong each frequency is attenuated depends on two factors: The entries of the system matrix and the numerical solver that is used. Thereby the overall strength of the attenuation determines the speed of convergence.

Local Mode Analysis - Concrete Example. In Figure 4.3 the attenuation behaviour of the standard Gauß-Seidel method for our prototype A is studied. It shows the development of the Fourier spectrum of the error for an increasing number of iterations. In order to make the attenuation behaviour more explicit, the standard Gauß-Seidel method was initialised in such a way that all frequencies contributed equally to the initial error (*white error spectrum*). The different spectra of the standard Gauß-Seidel method show the typical problem of non-hierarchical splitting-based iterative solvers [BHM00]: While *high frequency* components (small wavelength, local impact) are attenuated efficiently, *lower frequency* components (large wavelength, global impact) remain almost un-dampened. In practice, this smoothing behaviour with respect to the error results in a significant deterioration of the convergence rate already after a few iterations. Then, only low frequency components remain that only allow for a very slow attenuation by the underlying relaxation scheme. Apart from the typical problem of weak attenuation, the different spectra show also the dependency of the Gauß-Seidel solver on the order of traversal. This is reflected in the asymmetry of the resulting error spectra with respects to the two frequency axes.

Fig. 4.3: Logarithmic Fourier spectra of the numerical error in the u-component of the optic flow for our prototype A. Development for an increasing number of standard Gauß-Seidel iterations. Lower frequencies are located in the centre of the image, higher frequencies at the boundaries. **(a) Outer Left:** Initialisation with equally distributed error frequencies. **(b) Centre Left:** 1 iteration. **(c) Centre Right:** 10 iterations. **(d) Outer Right:** 100 iterations.

Basic Idea - The Linear Bidirectional Two-Grid Cycle. In order to overcome this problem bidirectional multigrid methods [Bra77, BHM00, Hac85, TOS01, Wes92] are based on a sophisticated strategy. They make use of correction steps that compute the *error* (not a coarser version of the fine grid solution) on a coarser grid. Thus, lower frequency components of the error reappear as higher ones and allow for an efficient attenuation with standard iterative methods. In the following we explain this strategy in detail by the example of a basic bidirectional two-grid cycle.

I) *Presmoothing Relaxation.* The bidirectional two-grid cycle starts by performing ν_1 iterations with a basic iterative solver (e.g. the standard Gauß–Seidel method). This is the so-called *presmoothing relaxation step*, where high frequency components of the error are removed. In Figure 4.4 the corresponding Fourier spectra for our prototype A are shown. As one can see, the high frequency components are attenuated very well. If we denote the correct solution by $\mathbf{x^h}$ and the result after these ν_1 iterations by $\tilde{\mathbf{x}}^h$, the current numerical error is given by

$$\mathbf{e^h} = \mathbf{x^h} - \tilde{\mathbf{x}}^h. \tag{4.45}$$

In the following this numerical error is in the focus of our computational efforts.
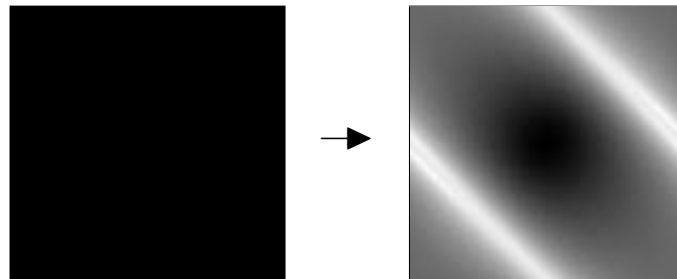
Fig. 4.4: Logarithmic Fourier spectra of the numerical error for the standard Gauß-Seidel method (prototype A). **(a) Left:** Before the presmoothing relaxation step. **(b) Right:** After the presmoothing relaxation step (2 iterations).

II) *Coarse Grid Transfer.* Evidently, one is interested in finding $\mathbf{e^h}$ in order to correct the approximated solution $\mathbf{\tilde{x}^h}$. Although $\mathbf{e^h}$ cannot be computed directly, the linearity of $A^\mathbf{h}$ allows its computation via

$$A^\mathbf{h}\mathbf{e^h} = A^\mathbf{h}(\mathbf{x^h} - \mathbf{\tilde{x}^h}) = A^\mathbf{h}\mathbf{x^h} - A^\mathbf{h}\mathbf{\tilde{x}^h} = \mathbf{b^h} - A^\mathbf{h}\mathbf{\tilde{x}^h} = \mathbf{r^h}, \qquad (4.46)$$

where $\mathbf{r^h} = \mathbf{b^h} - A^\mathbf{h}\mathbf{\tilde{x}^h}$ is called *residual*. Since most of the high frequencies of the error have already been attenuated, we can speed up the computation by solving this *residual equation system* at a coarser resolution:

$$A^\mathbf{h}\mathbf{e^h} = \mathbf{r^h} \quad \rightarrow \quad A^\mathbf{H}\mathbf{e^H} = \mathbf{r^H}. \qquad (4.47)$$

One should note that at this point, a transfer of the original equation system to a coarser grid makes no sense: Unlike the error, the solution very probably contains (desired) high frequency components. A restriction of these components would only deteriorate the solution (aliasing).

III) *Coarse Grid Solution.* Since the remaining *low* frequency components of the presmoothing relaxation step reappear on the coarser grid as *higher* frequencies they can now be (more) efficiently attenuated by applying the basic iterative solver.

IV) *Fine Grid Transfer.* After we have solved the residual equation system on the coarse grid, we have to transfer the solution back to the fine grid:

$$\mathbf{e^H} \quad \rightarrow \mathbf{\tilde{e}^h}. \qquad (4.48)$$

Here, the interpolated coarse grid solution is denoted by $\mathbf{\tilde{e}^h}$, since it is only an approximation to the desired correction $\mathbf{e^h}$.

V) *Coarse Grid Correction.* On the fine grid this interpolated solution can then be used to correct our approximative result from the presmoothing relaxation step:

$$\mathbf{\tilde{x}^h_{new}} = \mathbf{\tilde{x}^h} + \mathbf{\tilde{e}^h}. \qquad (4.49)$$

As one can see from the corresponding Fourier spectra for our prototype A in Figure 4.5, this correction allows a significant reduction of the lower frequency parts. However, one can also observe that the interpolation of the error introduces some new high frequency components.
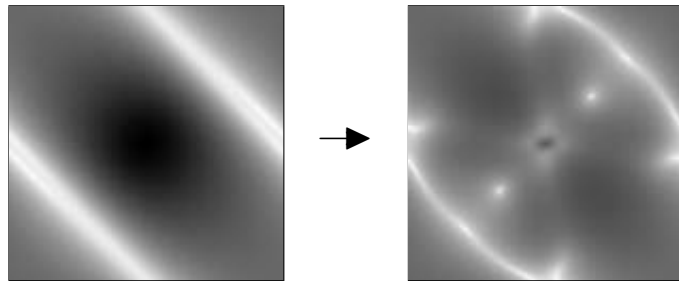
Fig. 4.5: Logarithmic Fourier spectra of the numerical error for the standard Gauß-Seidel method (prototype A). **(a) Left:** Before the coarse grid correction step. **(b) Right:** After the coarse grid correction step.

VI) *Postsmoothing Relaxation.* Finally, in order to eliminate these new high frequency components a so-called *postsmoothing relaxation step* is performed. Analogously to the presmoothing relaxation step this step is based on the application of ν_2 iterations of our basic iterative solver. The corresponding Fourier spectra for our prototype A are depicted in Figure 4.6. As expected they show a strong decrease of the high frequency components of the error.
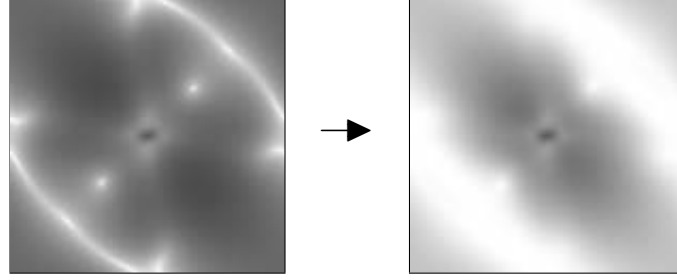
Fig. 4.6: Logarithmic Fourier spectra of the numerical error for the standard Gauß-Seidel method (prototype A). **(a) Left:** Before the postsmoothing relaxation step. **(b) Right:** After the postsmoothing relaxation step (2 iterations).

This six steps form the *bidirectional two-grid cycle* - the basic building block for the design of more advanced bidirectional multigrid methods. However, before we explain such methods in detail, let us discuss how to chose the intergrid transfer operators and the coarse grid equation system in the two-grid case.

Intergrid Transfer Operators. As in the unidirectional case we have to define a suitable pair of intergrid transfer operators for traversing the grids.

- *Restriction Operator.* In order to transfer the right hand side of the residual equation system from the fine to the coarse grid, a restriction operator $R^{\mathbf{h}\to\mathbf{H}}$ is required:

$$\mathbf{r}^{\mathbf{H}} = R^{\mathbf{h}\to\mathbf{H}}\mathbf{r}^{\mathbf{h}}. \tag{4.50}$$

  For this purpose the injection operator (pure resampling), the full weighting operator [BHM00] and the operator for area based averaging [BWF+05] are a good choice.

- *Prolongation Operator.* Moreover, we need a prolongation operator $P^{\mathbf{H}\to\mathbf{h}}$ to transfer the coarse grid correction back to the fine grid:

$$\tilde{\mathbf{e}}^{\mathbf{h}} = P^{\mathbf{H}\to\mathbf{h}}\mathbf{e}^{\mathbf{H}}. \tag{4.51}$$

  Also in this case we can use the same operators as proposed in the unidirectional case: constant/linear interpolation [BHM00] or area based interpolation [BWF+05].

However, in the bidirectional case, the transfer operator pair has to fulfil a certain condition [Hac85, Wes92]. If we denote by $\mathrm{order}(P)$ the maximum degree of a polynomial that can still be interpolated exactly be the prolongation operator P and if m stands for the order of the partial differential equations that shall be solved, then this condition reads:

$$\mathrm{order}(\mathbf{P}^{\mathbf{H}\to\mathbf{h}}) + \mathrm{order}((\mathbf{R}^{\mathbf{h}\to\mathbf{H}})^{\top}) > m - 2. \tag{4.52}$$

As we will see in our concrete example, this condition can be fulfilled by area-based interpolation and area-based averaging.

Coarse Grid Equation System. In order to set up the residual equation system for the coarse grid correction $A^{\mathbf{H}}\mathbf{e}^{\mathbf{H}} = \mathbf{r}^{\mathbf{H}}$, not only a restriction operator for the residual is needed but also a suitable coarse grid representation of matrix $A^{\mathbf{h}}$. Analogously to the unidirectional case, this coarse grid matrix can be obtained by one of the following three strategies:

- Problem Based Coarse Grid Approximation (PCA)

- Discretisation Coarse Grid Approximation (DCA)

- Galerkin Coarse Grid Approximation (GCA)

One should note that the approximation quality of the coarse grid operator $A^{\mathbf{H}}$ with respect to its fine grid counterpart $A^{\mathbf{h}}$ is much more important for bidirectional multigrid methods than for unidirectional ones. This is reflected in the fact that simple approaches based on the problem based coarse grid approximation (PCA) do not converge properly in most of the cases [Gla84, Enk87, ER03]. However, this poor behaviour with respect to a PCA does not mean that bidirectional multigrid methods are not suitable to solve systems of equations that are derived from optic flow methods as stated in [BAK91].

Hierarchical Application. In our explanation of the basic two-grid cycle the question of an appropriate solver for the residual equation system on the coarse grid was left open. A very elegant and efficient strategy in this context is the use of a third grid, that provides a correction for the second grid itself. Such a hierarchical application of the presented two-grid cycle is called *V–cycle*. One may also think of performing a second, additional correction step per level. This yields the so-called *W–cycle* that has better convergence properties at the expense of slightly increased computational costs. Examples for the grid traversal strategies of V and W–cycles are given in Figure 4.7.

One should note that in general multiple of these advanced cycles are required to reach the desired accuracy. This can also be seen from Figure 4.8 that shows the development of the error frequencies over three V-cycles: The low and middle frequency components that
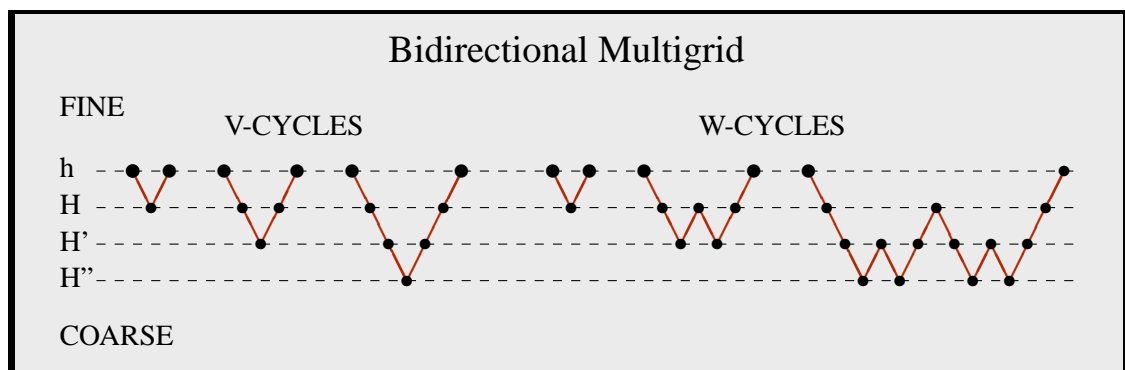
Fig. 4.7: Schematic drawing of bidirectional strategies (V– and W-cycles) for two, three and four levels. Iterations with the basic solver on the original problem are marked with large black dots, while iterations on residual problems are symbolised by small black dots.
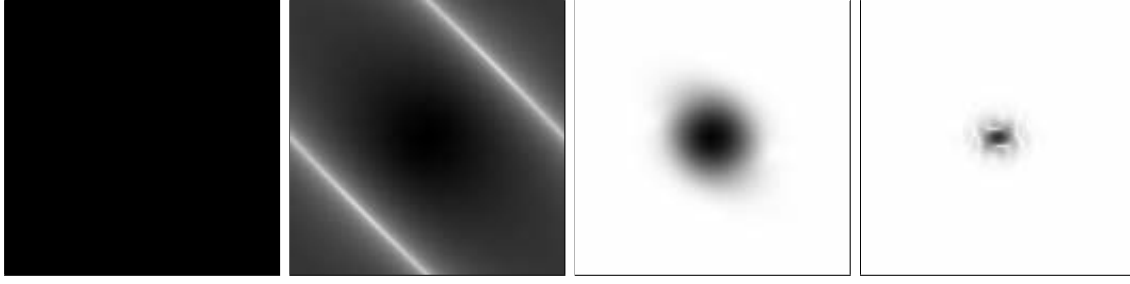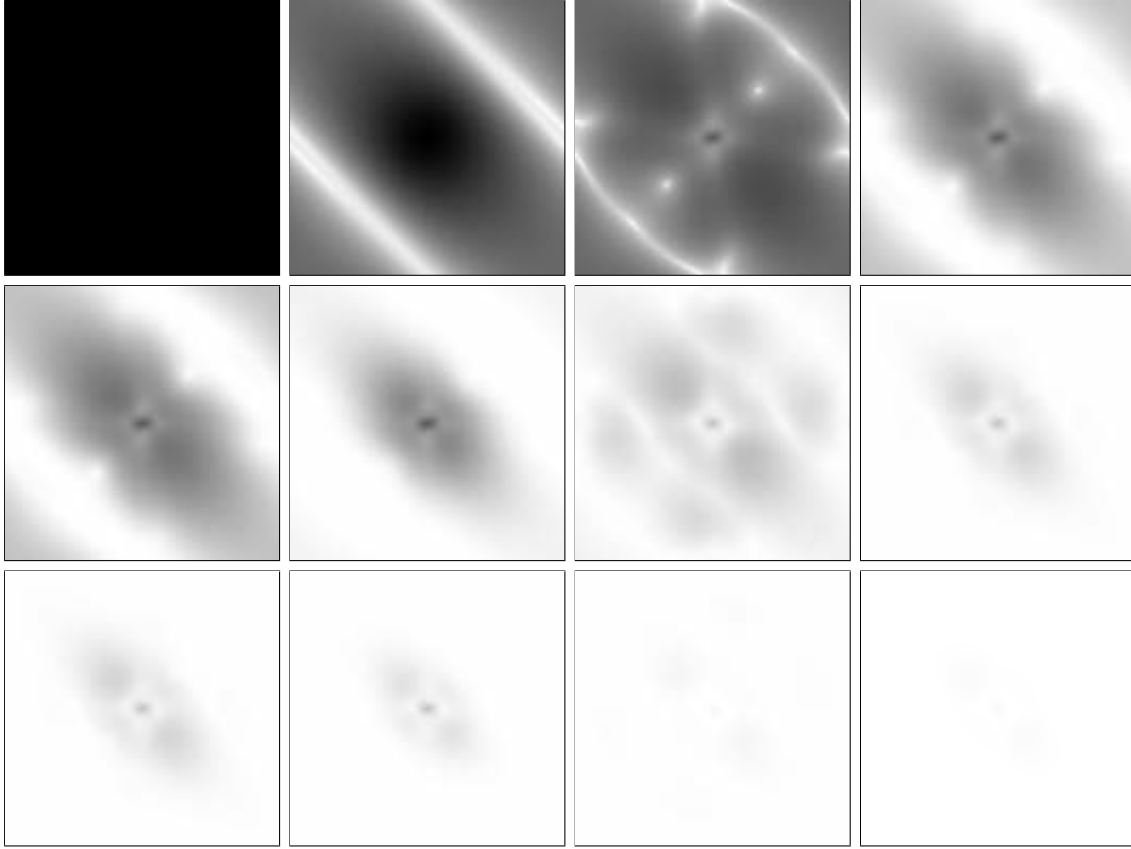
Fig. 4.8: Logarithmic Fourier spectra of the numerical error in the u-component of the optic flow for our prototype A. Error reduction over three V-cycles. **From Top to Bottom:** First cycle, second cycle and third cycle. **From Left to Right:** Initialisation, presmoothing relaxation step, coarse grid correction step, postsmoothing relaxation step.

remain after the first cycle are almost completely attenuated by the second one. After three cycles even these frequency components vanish. This in turn corresponds to a decrease of the Euclidean norm of the error of *four orders of magnitude*.

Concrete Example. At this point let us investigate how the proposed numerical method can actually be used for solving the equation system associated our prototype A. To this end, we discuss how a such a bidirectional multigrid scheme based on a discretisation coarse grid approximation (DCA) approach can be derived. This requires four decisions:

I) *Basic Solver*. Firstly, we have to select a basic iterative solver. In this context one should note that in contrast to the unidirectional case, one is only interested in the attenuation properties of the solver with respect to the *higher frequency parts*. The lower frequency parts are *automatically covered by construction* of the multigrid algorithm. This excludes advanced non-hierarchical solvers such as the SOR method that may have the better overall performance but cannot dampen high frequencies as efficient as the Gauß-Seidel method. For our concrete example we select once again a Gauß-Seidel variant: The Gauß-Seidel method with coupled point relaxation (CPR).

II) *Intergrid Transfer Operators.* Secondly, a suitable pair of intergrid transfer operators has to be defined. Since the order of our differential equation is 2, the accuracy condition (4.52) tells us that at least one of the two operators (prolongation, transposed restriction) has to be of order 1. Therefore, *area based interpolation* (order 0) and *area based averaging* (transposed order 1) are a suitable choice [BWF$^+$05, BWKS06].

III) *Coarse Grid Cell Size.* Thirdly, a new coarse grid size $\mathbf{H}$ has to be chosen. In contrast to all other bidirectional multigrid approaches for variational optic flow techniques with homogeneous or image-driven regularisation [Gla84, Ter86, Enk87, ZSL97, BWF$^+$03, ER03] we decided *not to limit* ourselves to image sizes, where N_{x_1} and N_{x_2} are a power of 2. This makes our approach much more flexible, in particular with respect to the fact that images of any size may occur. At this point one should note that extending smaller images with zeros (*zero padding*) is not a good solution: Such a proceeding yields significantly different results at the boundaries.

As in the unidirectional case we propose the following computation of $H_{x_1} \times H_{x_2}$. Let $N_{x_1}^{\mathbf{h}}$ and $N_{x_2}^{\mathbf{h}}$ be the number of cells on the fine grid in x and y direction. Then the new cell size is given by

$$H_{x_1} := h_{x_1} \frac{N_{x_1}^{\mathbf{h}}}{N_{x_1}^{\mathbf{H}}} \qquad\qquad H_{x_2} := h_{x_2} \frac{N_{x_2}^{\mathbf{h}}}{N_{x_2}^{\mathbf{H}}} \qquad (4.53)$$

with $N_{x_1}^H = \lceil N_{x_1}^{\mathbf{h}}/2 \rceil$ and $N_{x_2}^{\mathbf{H}} = \lceil N_{x_2}^{\mathbf{h}}/2 \rceil$, where $\lceil z \rceil$ is the smallest integer number m with $m \geq z$. Thus the total number of cells at the coarser grid is $N^{\mathbf{H}} = N_{x_1}^{\mathbf{H}} N_{x_2}^{\mathbf{H}}$. One should note that such a non-dyadic approach becomes only possible, because our area-based intergrid transfer operators are non-dyadic be construction. The frequently used *full weighting operator* [BHM00], for instance, cannot be used for restricting arbitrary image sizes.

IV) *Coarse Grid Equation System.* Finally, we have to set up the the *residual equation system* for the coarse grid. To this end, we follow the discretisation coarse grid approximation approach (DCA) and rediscretise all entries of the discrete Euler–Lagrange equations that belong to the matrix $A^{\mathbf{h}}$. This yields the coarse grid equation system

$$\left[J_{11}^{\mathbf{H}}\right]_{i,j} \left[e_1^{\mathbf{h}}\right]_{i,j} + \left[J_{12}^{\mathbf{H}}\right]_{i,j} \left[e_2^{\mathbf{H}}\right]_{i,j} - \alpha \sum_{l \in \{x_1, x_2\}} \sum_{\tilde{i}, \tilde{j} \in \mathcal{N}_l(i,j)} \frac{\left(\left[e_1^{\mathbf{H}}\right]_{\tilde{i},\tilde{j}} - \left[e_1^{\mathbf{H}}\right]_{i,j}\right)}{H_l^2} = \left[r_1^{\mathbf{H}}\right]_{i,j},$$

$$(4.54)$$

$$\left[J_{12}^{\mathbf{h}}\right]_{i,j} \left[e_1^{\mathbf{h}}\right]_{i,j} + \left[J_{22}^{\mathbf{H}}\right]_{i,j} \left[e_2^{\mathbf{H}}\right]_{i,j} - \alpha \sum_{l \in \{x_1, x_2\}} \sum_{\tilde{i}, \tilde{j} \in \mathcal{N}_l(i,j)} \frac{\left(\left[e_2^{\mathbf{H}}\right]_{\tilde{i},\tilde{j}} - \left[e_2^{\mathbf{h}}\right]_{i,j}\right)}{H_l^2} = \left[r_2^{\mathbf{H}}\right]_{i,j},$$

$$(4.55)$$

for $i = 1, ..., N_{x_1}^{\mathbf{H}}$ and $j = 1, ..., N_{x_2}^{\mathbf{H}}$, where $\left[e_1^{\mathbf{H}}\right]_{i,j}$ and $\left[e_2^{\mathbf{H}}\right]_{i,j}$ are the new unknowns (coarse grid error) and $\left[r_1^{\mathbf{H}}\right]_{i,j}$ and $\left[r_2^{\mathbf{H}}\right]_{i,j}$ are the components of the restricted residual. As in the unidirectional case, the entries of the coarse grid motion tensors $\left[J^{\mathbf{H}}\right]_{i,j}$ and the coarse grid diffusion tensors $\left[T^{\mathbf{H}}\right]_{i,j}$ are obtained via channelwise restriction.

In this context one should note that in the optic flow literature the problem based coarse grid approximation approach (PCA) is by far most frequently used [Gla84, Ter86, Enk87, ZSL97, ER03]. However, as we have discussed already in the unidirectional case, the resampling of the data (image sequence/derivatives of the image sequence) is definitely not the best solution. It is therefore not surprising that due to the inaccurate representation of the coarse grid equation systems in the case of the PCA, convergence problems arise if the number of levels is increased [Gla84, ER03]. As shown in [BWF⁺03] this is not the case for the discretisation coarse grid approximation (DCA).

This example shows that also the design of bidirectional methods can be simplified significantly if our motion and diffusion tensor notation is used. In combination with the discretisation coarse grid approximation approach (DCA) this notation not only allows for a systematic design of variational models but also for a systematic construction of efficient numerical schemes for their minimisation.

Full Multigrid. So far we have strictly separated the idea of *hierarchical initialisation* (unidirectional multigrid methods) and *coarse grid correction* (bidirectional multigrid methods). However, it is not difficult to see that both strategies are complementary: While coarse-to-fine strategies could benefit from a better solver at each resolution level, V- and W-cycles could benefit from a better initialisation. Consequently, both concepts should be combined to obtain optimal convergence. If this is done, the class of bidirectional multigrid methods with the best performance is obtained: *full multigrid*. In the ideal case such methods have linear complexity[3] $O(n)$, i.e. the computational costs depends linearly on the number of unknowns. Evidently, from a theoretical point of view, this is optimal.

An example for a full multigrid method with four levels is given in Figure 4.9. As one can see the original equation system is refined step by step. Thereby at each resolution level, two W-cycles are used as solvers.
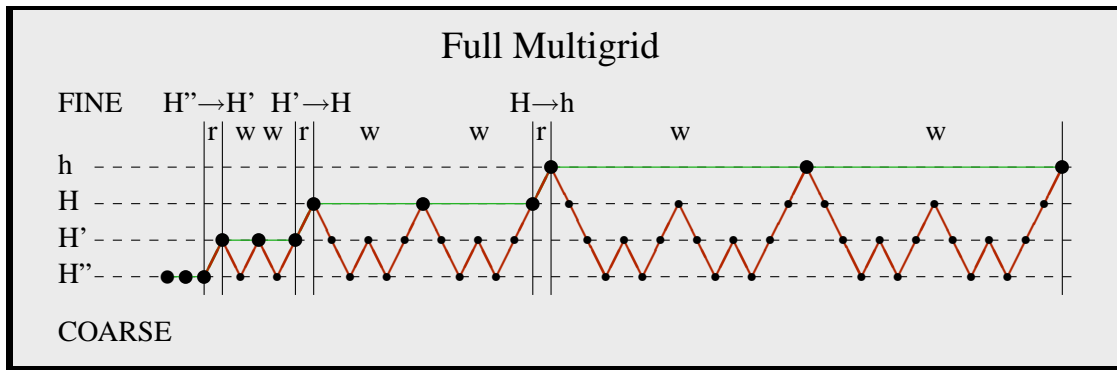
Fig. 4.9: Schematic drawing of a full multigrid strategy, Starting from a very coarse variant of the original equation system, the solution is refined step by step. The corresponding refinements steps are marked with 'r'. At each resolution level two W-cycles are used as solver. Each of these cycles is marked with 'w'. Iterations with the basic solver on the original problem are represented with large black dots, while iterations on residual problems are symbolised by small black dots.

---

3. This asymptotical complexity refers to simple model problems such as the Poisson equation [TOS01].

4.2   The Nonlinear Case I + II

After we have shown how to design efficient numerical schemes for the linear case, let us now extend this systematic approach to the nonlinear cases I and II. To this end, we focus on *inexact* variants of *Lagged-Diffusivity* methods [CM99, Vog02] that are known in the literature as techniques with *frozen coefficients* [FSHW04]. As in the linear case these basic solvers are first discussed on a single grid before they are used for the construction of efficient hierarchical methods. Thereby we stick to our previous strategy for presenting the numerical schemes: First, we explain their basic idea in a rather general way. Then, we show by a concrete example – our prototype F – how these schemes can actually be applied in the context of variational optic flow methods.

General Problem.   Let us start by giving a general definition of our problem. In the following we are interested in solving a *nonlinear* equation system of type

$$A^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}}) = \mathbf{b}^{\mathbf{h}}, \tag{4.56}$$

where $A^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})$ is a nonlinear operator and $\mathbf{b}^{\mathbf{h}}$ is the right hand side. Thereby $A^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})$ can be decomposed into

$$A^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}}) = B^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})\,\mathbf{x}^{\mathbf{h}} + \mathbf{c}^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}}) \tag{4.57}$$

where $B^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})$ and $\mathbf{c}^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})$ are nonlinear operators such that for each $\mathbf{x}^{\mathbf{h}}$ the value $B^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})$ is a *symmetric and positive definite* $2N^{\mathbf{h}} \times 2N^{\mathbf{h}}$ matrix with $O(N^{\mathbf{h}})$ non-zero entries, while $\mathbf{c}^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})$ is a vector of size $2N^{\mathbf{h}}$. As we have discussed in 3.3.3.2 and 3.3.3.3 such linear systems are obtained for our prototypes D-G. Please recall that the convexity of the underlying energy functionals guarantees a unique solution of this nonlinear system.

Concrete Example.   In our concrete example we are interested in solving the discrete Euler–Lagrange equations that are associated with our prototype F (method of Bruhn *et al.*). To this end, let us first identify the different parts of the corresponding nonlinear equation system in (3.42) with those of our general problem in (4.56)–(4.57). This can be done as follows:

$$\underbrace{\left(\begin{pmatrix} F_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}})J_{11}^{\mathbf{h}} & F_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}})J_{12}^{\mathbf{h}} \\ F_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}})J_{12}^{\mathbf{h}} & F_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}})J_{22}^{\mathbf{h}} \end{pmatrix} - \alpha \begin{pmatrix} L_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}}) & 0 \\ 0 & L_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}}) \end{pmatrix} \right)}_{B^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})} \underbrace{\begin{pmatrix} \mathbf{u}_1^{\mathbf{h}} \\ \mathbf{u}_2^{\mathbf{h}} \end{pmatrix}}_{\mathbf{x}^{\mathbf{h}}}$$

$$+ \underbrace{\begin{pmatrix} F_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}})\,\mathbf{j}_{13}^{\mathbf{h}} \\ F_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}})\,\mathbf{j}_{23}^{\mathbf{h}} \end{pmatrix}}_{\mathbf{c}^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})} = \underbrace{\begin{pmatrix} 0 \\ 0 \end{pmatrix}}_{\mathbf{b}^{\mathbf{h}}}. \tag{4.58}$$

One should note that our prototype F represents the nonlinear case II. For the sake of completeness, let us therefore also derive the assignment of the different operators for the (simpler) nonlinear case I. They are given by

$$\underbrace{\left(\begin{pmatrix} J_{11}^{\mathbf{h}} & J_{12}^{\mathbf{h}} \\ J_{12}^{\mathbf{h}} & J_{22}^{\mathbf{h}} \end{pmatrix} - \alpha \begin{pmatrix} L_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}}) & 0 \\ 0 & L_{\mathrm{NL}}^{\mathbf{h}}(\mathbf{u}_1^{\mathbf{h}}, \mathbf{u}_2^{\mathbf{h}}) \end{pmatrix} \right)}_{B^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})} \underbrace{\begin{pmatrix} \mathbf{u}_1^{\mathbf{h}} \\ \mathbf{u}_2^{\mathbf{h}} \end{pmatrix}}_{\mathbf{x}^{\mathbf{h}}} + \underbrace{\begin{pmatrix} 0 \\ 0 \end{pmatrix}}_{\mathbf{c}^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})} = \underbrace{\begin{pmatrix} -\mathbf{j}_{13}^{\mathbf{h}} \\ -\mathbf{j}_{23}^{\mathbf{h}} \end{pmatrix}}_{\mathbf{b}^{\mathbf{h}}} .$$

$$(4.59)$$

As in the linear case we have to reformulate this equation system in point-based notation in order to make the effect of the discrete nonlinear differential operator $L_{\mathrm{NL}}^{\mathbf{h}}$ and the discrete nonlinear operator F_{NL} on $\mathbf{u}_1^{\mathbf{h}}$ and $\mathbf{u}_2^{\mathbf{h}}$ explicit. Thus, we obtain

$$\left[\Psi_{\mathrm{D}}'^{\mathbf{h}} \right]_{i,j} \left[J_{11}^{\mathbf{h}} \right]_{i,j} \left[u_1^{\mathbf{h}} \right]_{i,j} + \left[\Psi_{\mathrm{D}}'^{\mathbf{h}} \right]_{i,j} \left[J_{12}^{\mathbf{h}} \right]_{i,j} \left[u_2^{\mathbf{h}} \right]_{i,j} + \left[\Psi_{\mathrm{D}}'^{\mathbf{h}} \right]_{i,j} \left[J_{13}^{\mathbf{h}} \right]_{i,j}$$

$$- \alpha \sum_{l \in \{x_1, x_2\}} \sum_{\tilde{i}, \tilde{j} \in \mathcal{N}_l(i,j)} \frac{\left(\left[\Psi_{\mathrm{S}}'^{\mathbf{h}} \right]_{\tilde{i}, \tilde{j}} + \left[\Psi_{\mathrm{S}}'^{\mathbf{h}} \right]_{i,j} \right) \left(\left[u_1^{\mathbf{h}} \right]_{\tilde{i}, \tilde{j}} - \left[u_1^{\mathbf{h}} \right]_{i,j} \right)}{2} \frac{1}{h_l^2} = 0 \quad (4.60)$$

$$\left[\Psi_{\mathrm{D}}'^{\mathbf{h}} \right]_{i,j} \left[J_{12}^{\mathbf{h}} \right]_{i,j} \left[u_1^{\mathbf{h}} \right]_{i,j} + \left[\Psi_{\mathrm{D}}'^{\mathbf{h}} \right]_{i,j} \left[J_{22}^{\mathbf{h}} \right]_{i,j} \left[u_2^{\mathbf{h}} \right]_{i,j} + \left[\Psi_{\mathrm{D}}'^{\mathbf{h}} \right]_{i,j} \left[J_{13}^{\mathbf{h}} \right]_{i,j}$$

$$- \alpha \sum_{l \in \{x_1, x_2\}} \sum_{\tilde{i}, \tilde{j} \in \mathcal{N}_l(i,j)} \frac{\left(\left[\Psi_{\mathrm{S}}'^{\mathbf{h}} \right]_{\tilde{i}, \tilde{j}} + \left[\Psi_{\mathrm{S}}'^{\mathbf{h}} \right]_{i,j} \right) \left(\left[u_2^{\mathbf{h}} \right]_{\tilde{i}, \tilde{j}} - \left[u_2^{\mathbf{h}} \right]_{i,j} \right)}{2} \frac{1}{h_l^2} = 0 \quad (4.61)$$

for $i = 1, ..., N_{x_1}^{\mathbf{h}}$ and $j = 1, ..., N_{x_2}^{\mathbf{h}}$. Here, $\left[\Psi_{\mathrm{D}}'^{\mathbf{h}} \right]_{i,j}$ and $\left[\Psi_{\mathrm{S}}'^{\mathbf{h}} \right]_{i,j}$ denote the functions $\Psi_{\mathrm{D}}'(s^2)$ and $\Psi_{\mathrm{S}}'(s^2)$ evaluated at pixel (i, j), i.e.

$$\left[\Psi_{\mathrm{D}}'^{\mathbf{h}} \right]_{i,j} = \frac{1}{2 \sqrt{ \left(\left[u_1^{\mathbf{h}} \right]_{i,j}, \left[u_2^{\mathbf{h}} \right]_{i,j}, 1 \right)^{\top} (J^{\mathbf{h}})_{i,j} \left(\left[u_1^{\mathbf{h}} \right]_{i,j}, \left[u_2^{\mathbf{h}} \right]_{i,j}, 1 \right) + \epsilon_{\mathrm{D}}^2 }} , \quad (4.62)$$

$$\left[\Psi_{\mathrm{S}}'^{\mathbf{h}} \right]_{i,j} = \frac{1}{2 \sqrt{ |D^{2,\mathbf{h}}(\left[u_1^{\mathbf{h}} \right]_{i,j})|^2 + |D^{2,\mathbf{h}}(\left[u_2^{\mathbf{h}} \right]_{i,j})|^2 + \epsilon_{\mathrm{S}}^2 }} , \quad (4.63)$$

where $|D^{2,\mathbf{h}}(\left[z^{\mathbf{h}} \right]_{i,j})|$ denotes the gradient magnitude operator as defined in Table 3.1.

In this context one should note that the nonlinearity of the preceding system is now hidden in the expressions $\left[\Psi_{\mathrm{D}}'^{\mathbf{h}} \right]_{i,j}$ and $\left[\Psi_{\mathrm{S}}'^{\mathbf{h}} \right]_{i,j}$ that actually depend on $\left[u_1^{\mathbf{h}} \right]_{i,j}$ and $\left[u_2^{\mathbf{h}} \right]_{i,j}$. Thus it is not surprising that both nonlinear expressions become 1 if the nonquadratic penalisers $\Psi_{\mathrm{D}}(s^2) = \sqrt{s^2 + \epsilon_{\mathrm{D}}^2}$ and $\Psi_{\mathrm{S}}(s^2) = \sqrt{s^2 + \epsilon_{\mathrm{S}}^2}$ are replaced by their quadratic counterparts $\Psi_{\mathrm{D}}(s^2) = s^2$ and $\Psi_{\mathrm{D}}(s^2) = s^2$. In this case, the associated nonlinear system of equations basically simplifies – apart from the different motion tensor – to the linear one of our prototype A (4.3–4.4).

4.2.1 The Lagged-Diffusivity Method

Basic Idea. A general approach for solving the nonlinear system of equations in (4.56) is its decomposition into a series of linear problems and the use of standard linear techniques such as the Gauß-Seidel method. This can be accomplished by means of the so-called

Lagged-Diffusivity or *Kačanov* method [KNPS68, FKN73, CM99]. By introducing a fixed point iteration of the form

$$\mathbf{x}^{\mathbf{h},\,k+1} = \left(B^{\mathbf{h}}(\mathbf{x}^{\mathbf{h},\,k})\right)^{-1}\left(\mathbf{b}^{\mathbf{h}} - \mathbf{c}^{\mathbf{h}}(\mathbf{x}^{\mathbf{h},\,k})\right) \tag{4.64}$$

this method exploits both the decomposition of $A^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})$ into $B^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})\,\mathbf{x}^{\mathbf{h}} + \mathbf{c}^{\mathbf{h}}(\mathbf{x}^{\mathbf{h},\,k})$ and the special properties of $B^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})$ and $\mathbf{c}^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})$ with respect to a fixed argument $\mathbf{x}^{\mathbf{h}}$. The basic idea is thereby to use the old (lagged) solution $\mathbf{x}^{\mathbf{h},\,k}$ for evaluating the nonlinear operators $B^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})$ and $\mathbf{c}^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})$. Thus, a linear system remains to be solved for *each* iteration step:

$$B^{\mathbf{h}}(\mathbf{x}^{\mathbf{h},\,k})\,\mathbf{x}^{\mathbf{h},\,k+1} = \left(\mathbf{b}^{\mathbf{h}} - \mathbf{c}^{\mathbf{h}}(\mathbf{x}^{\mathbf{h},\,k})\right). \tag{4.65}$$

This task, however, is straightforward: Since the corresponding system matrix $B^{\mathbf{h}}(\mathbf{x}^{\mathbf{h},\,k})$ is symmetric and positive definite, each of these steps comes down to our general problem for the linear case (4.1). For this problem, in turn, we have already proposed a variety of efficient methods.

Concrete Example. Let us now discuss by our concrete example (prototype F – the method of Bruhn *et al.*) how the Lagged-Diffusivity method can be used to solve the Euler-Lagrange equations for the nonlinear cases I and II. This is done in two steps:

I) *The Fixed Point Iteration*. Firstly, we have to derive the linear system of equations that has to be solved in *each* step of the fixed point iteration (4.65). For our prototype F this system is given by

$$\left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}\left[J_{11}^{\mathbf{h}}\right]_{i,j}\left[u_1^{\mathbf{h}}\right]_{i,j}^{k+1} + \left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}\left[J_{12}^{\mathbf{h}}\right]_{i,j}\left[u_2^{\mathbf{h}}\right]_{i,j}^{k+1} + \left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}\left[J_{13}^{\mathbf{h}}\right]_{i,j}$$

$$-\alpha\sum_{l\in\{x_1,x_2\}}\sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)}\frac{\left(\left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k} + \left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{i,j}^{k}\right)}{2}\frac{\left(\left[u_1^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k+1} - \left[u_1^{\mathbf{h}}\right]_{i,j}^{k+1}\right)}{h_l^2} = 0 \tag{4.66}$$

$$\left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}\left[J_{12}^{\mathbf{h}}\right]_{i,j}\left[u_1^{\mathbf{h}}\right]_{i,j}^{k+1} + \left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}\left[J_{22}^{\mathbf{h}}\right]_{i,j}\left[u_2^{\mathbf{h}}\right]_{i,j}^{k+1} + \left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}\left[J_{23}^{\mathbf{h}}\right]_{i,j}$$

$$-\alpha\sum_{l\in\{x_1,x_2\}}\sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)}\frac{\left(\left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k} + \left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{i,j}^{k}\right)}{2}\frac{\left(\left[u_2^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k+1} - \left[u_2^{\mathbf{h}}\right]_{i,j}^{k+1}\right)}{h_l^2} = 0 \tag{4.67}$$

for $i=1, ..., N_{x_1}^{\mathbf{h}}$ and $j=1, ..., N_{x_2}^{\mathbf{h}}$. In order to compute the $2N^{\mathbf{h}}$ unknowns $\left[u_1^{\mathbf{h}}\right]_{i,j}^{k+1}$ and $\left[u_2^{\mathbf{h}}\right]_{i,j}^{k+1}$, the originally nonlinear expressions $\left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}$ and $\left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{i,j}$ are kept fixed, i.e. they are evaluated at the old time step k. Therefore they are denoted by $\left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}^{k}$ and $\left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{i,j}^{k}$.

II) *The Basic Solver.* Secondly, we have to select a numerical method to solve each of these linear equation systems given by (4.66)–(4.67). In order to keep things simple we choose the Gauß-Seidel method with coupled point relaxation (CPR). However, one could also use any other method for the linear case. If we denote by n the iteration index associated to our Gauß-Seidel solver, the corresponding iteration instructions in point-notation read

$$
\begin{pmatrix} \left[u_1^{\mathbf{h}}\right]_{i,j}^{k+1,n+1} \\ \left[u_2^{\mathbf{h}}\right]_{i,j}^{k+1,n+1} \end{pmatrix} = \begin{pmatrix} \left[K_{11}^{\mathbf{h}}\right]_{i,j}^{k} & \left[K_{12}^{\mathbf{h}}\right]_{i,j}^{k} \\ \left[K_{12}^{\mathbf{h}}\right]_{i,j}^{k} & \left[K_{22}^{\mathbf{h}}\right]_{i,j}^{k} \end{pmatrix}^{-1} \begin{pmatrix} \left[b_1^{\mathbf{h}}\right]_{i,j}^{k+1,n} \\ \left[b_2^{\mathbf{h}}\right]_{i,j}^{k+1,n} \end{pmatrix} \tag{4.68}
$$

for $i = 1, ..., N_{x_1}^{\mathbf{h}}$ and $j = 1, ..., N_{x_2}^{\mathbf{h}}$ with the matrix entries

$$
\left[K_{11}^{\mathbf{h}}\right]_{i,j}^{k} = \left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}^{k} \left[J_{11}^{\mathbf{h}}\right]_{i,j} + \alpha \sum_{l \in \{x_1, x_2\}} \sum_{\tilde{i},\tilde{j} \in \mathcal{N}(i,j)} \frac{\left(\left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k} + \left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{i,j}^{k}\right)}{2h_l^2} \tag{4.69}
$$

$$
\left[K_{12}^{\mathbf{h}}\right]_{i,j}^{k} = \left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}^{k} \left[J_{12}^{\mathbf{h}}\right]_{i,j} \tag{4.70}
$$

$$
\left[K_{22}^{\mathbf{h}}\right]_{i,j}^{k} = \left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}^{k} \left[J_{22}^{\mathbf{h}}\right]_{i,j} + \alpha \sum_{l \in \{x_1, x_2\}} \sum_{\tilde{i},\tilde{j} \in \mathcal{N}(i,j)} \frac{\left(\left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k} + \left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{i,j}^{k}\right)}{2h_l^2} \tag{4.71}
$$

and right hand side

$$
\left[b_1^{\mathbf{h}}\right]_{i,j}^{k+1,n} = -\left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}^{k} \left[J_{13}^{\mathbf{h}}\right]_{i,j} + \alpha \sum_{l \in \{x_1, x_2\}} \sum_{\tilde{i},\tilde{j} \in \mathcal{N}_l^-(i,j)} \frac{\left(\left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k} + \left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{i,j}^{k}\right)}{2h_l^2} \left[u_1^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k+1,n+1}
$$
$$
+ \alpha \sum_{l \in \{x_1, x_2\}} \sum_{\tilde{i},\tilde{j} \in \mathcal{N}_l^+(i,j)} \frac{\left(\left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k} + \left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{i,j}^{k}\right)}{2h_l^2} \left[u_1^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k+1,n} \tag{4.72}
$$

$$
\left[b_2^{\mathbf{h}}\right]_{i,j}^{k+1,n} = -\left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}^{k} \left[J_{23}^{\mathbf{h}}\right]_{i,j} + \alpha \sum_{l \in \{x_1, x_2\}} \sum_{\tilde{i},\tilde{j} \in \mathcal{N}_l^-(i,j)} \frac{\left(\left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k} + \left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{i,j}^{k}\right)}{2h_l^2} \left[u_2^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k+1,n+1}
$$
$$
+ \alpha \sum_{l \in \{x_1, x_2\}} \sum_{\tilde{i},\tilde{j} \in \mathcal{N}_l^+(i,j)} \frac{\left(\left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k} + \left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{i,j}^{k}\right)}{2h_l^2} \left[u_2^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k+1,n}. \tag{4.73}
$$

This example shows the interaction between the actual fixed point iteration that keeps the nonlinearity fixed at each step and the basic iterative method that solves the resulting linear systems of equations.

4.2.2 The Inexact Lagged-Diffusivity Method

Basic Idea. In order to improve the overall performance of the Lagged-Diffusivity method, one may think of replacing the proposed Gauß-Seidel variant by a more efficient numerical solver, e.g. by a full multigrid scheme. However, this is not the best strategy to speed up the computation. More efficient, in particular in terms of computation costs, are so-called *inexact* variants of the Lagged-Diffusivity method. Instead of solving the linear systems of equations of type (4.65) with a very high precision, such techniques perform only a *few solver iterations* to obtain an approximate solution at each step [WHS$^+$01, Vog02] Evidently, such a proceeding leads to a slight deterioration of the convergence of the fixed point iteration itself. However, the number of additional fixed point steps is negligible compared to the significant reduction of computational costs for each equation system. Thus it is not surprising that *inexact* variants are much faster than any exact method that relies on solving the linear equation systems precisely. One should note the convergence of such inexact methods is not proven yet. Hints, however, can be found in [Axe97].

Frozen Coefficients. Special representatives for this class of Lagged-Diffusivity techniques are iterative methods with *frozen coefficients* [FSHW04]. Such methods are based on the idea of accelerating the fixed point iteration by updating the lagged nonlinear expressions as fast as possible. Consequently, these expressions are only kept fixed (frozen) for *one iteration* of the basic solver. The resulting strategy of combining fixed point and solver iterations can be interpreted in two ways: Either one may think of a "very inexact" Lagged-Diffusivity method that uses only one iteration to solve each equation system or one may think of a nonlinear iterative method that extends the idea of linear iterative solvers by an additional update step.

Concrete Example. Let us now investigate how the strategy of frozen coefficients can be applied for solving the nonlinear equation system associated to our prototype F (method of Bruhn *et al.*). In order to allow for a comparison to the exact Lagged-Diffusivity method in (4.66)–(4.67) and (4.68)–(4.73) we have selected once more the Gauß-Seidel method with coupled point relaxation (CPR). The corresponding iteration instructions read

$$
\begin{pmatrix} \left[u_1^{\mathbf{h}}\right]_{i,j}^{k+1} \\ \left[u_2^{\mathbf{h}}\right]_{i,j}^{k+1} \end{pmatrix} = \begin{pmatrix} \left[K_{11}^{\mathbf{h}}\right]_{i,j}^{k} & \left[K_{12}^{\mathbf{h}}\right]_{i,j}^{k} \\ \left[K_{12}^{\mathbf{h}}\right]_{i,j}^{k} & \left[K_{22}^{\mathbf{h}}\right]_{i,j}^{k} \end{pmatrix}^{-1} \begin{pmatrix} \left[b_1^{\mathbf{h}}\right]_{i,j}^{k} \\ \left[b_2^{\mathbf{h}}\right]_{i,j}^{k} \end{pmatrix}
\tag{4.74}
$$

for $i = 1, ..., N_{x_1}^{\mathbf{h}}$ and $j = 1, ..., N_{x_2}^{\mathbf{h}}$ with the matrix entries

$$
\left[K_{11}^{\mathbf{h}}\right]_{i,j}^{k} = \left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}^{k} \left[J_{11}^{\mathbf{h}}\right]_{i,j} + \alpha \sum_{l \in \{x_1, x_2\}} \sum_{\tilde{i},\tilde{j} \in \mathcal{N}(i,j)} \frac{\left(\left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k} + \left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{i,j}^{k}\right)}{2h_l^2}
\tag{4.75}
$$

$$
\left[K_{12}^{\mathbf{h}}\right]_{i,j}^{k} = \left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}^{k} \left[J_{12}^{\mathbf{h}}\right]_{i,j}
\tag{4.76}
$$

$$
\left[K_{22}^{\mathbf{h}}\right]_{i,j}^{k} = \left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}^{k} \left[J_{22}^{\mathbf{h}}\right]_{i,j} + \alpha \sum_{l \in \{x_1, x_2\}} \sum_{\tilde{i},\tilde{j} \in \mathcal{N}(i,j)} \frac{\left(\left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k} + \left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{i,j}^{k}\right)}{2h_l^2}
\tag{4.77}
$$

and right hand side

$$
\begin{aligned}
\left[b_1^{\mathbf{h}}\right]_{i,j}^{k} = -\left[\Psi_D'^{\mathbf{h}}\right]_{i,j}^{k}\left[J_{13}^{\mathbf{h}}\right]_{i,j} &+ \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l^-(i,j)} \frac{\left(\left[\Psi_S'^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k} + \left[\Psi_S'^{\mathbf{h}}\right]_{i,j}^{k}\right)}{2h_l^2}\left[u_1^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k+1} \\
&+ \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l^+(i,j)} \frac{\left(\left[\Psi_S'^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k} + \left[\Psi_S'^{\mathbf{h}}\right]_{i,j}^{k}\right)}{2h_l^2}\left[u_1^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k}
\end{aligned}
$$

(4.78)

$$
\begin{aligned}
\left[b_2^{\mathbf{h}}\right]_{i,j}^{k} = -\left[\Psi_D'^{\mathbf{h}}\right]_{i,j}^{k}\left[J_{23}^{\mathbf{h}}\right]_{i,j} &+ \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l^-(i,j)} \frac{\left(\left[\Psi_S'^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k} + \left[\Psi_S'^{\mathbf{h}}\right]_{i,j}^{k}\right)}{2h_l^2}\left[u_2^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k+1} \\
&+ \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l^+(i,j)} \frac{\left(\left[\Psi_S'^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k} + \left[\Psi_S'^{\mathbf{h}}\right]_{i,j}^{k}\right)}{2h_l^2}\left[u_2^{\mathbf{h}}\right]_{\tilde{i},\tilde{j}}^{k}
\end{aligned}
$$

(4.79)

As one can see, in contrast to the exact case only one iteration index (k) is sufficient. This is a consequence of the fact that the indices for the fixed point iterations and solver iterations coincide: The frozen coefficients $\left[\Psi_D'^{\mathbf{h}}\right]_{i,j}^{k}$ and $\left[\Psi_S'^{\mathbf{h}}\right]_{i,j}^{k}$ are now updated after *each* iteration. In this context one should note that this frequent evaluation may be very expensive compared to the iteration step itself: It may require more than 50% of the total computational costs.

4.2.3   Unidirectional Multigrid Methods

With inexact Lagged-Diffusivity methods we have proposed a class of non-hierarchical iterative solvers for our nonlinear cases I and II. However, as we have already discussed in the context of linear problems, such iterative single-grid methods are not well-suited with respect to a very local coupling of the unknowns (via the discrete differential operator). Therefore it is also desirable in the nonlinear cases I and II to develop multigrid schemes that speed up the computation. As in the linear case we start with the simplest multigrid strategy – the multigrid strategy that is based on a coarse-to-fine initialisation of the results: *unidirectional multigrid*.

Basic Idea. In fact, there is no reason why the idea of hierarchical initialisation should not work for a nonlinear problem. However, one has to be more careful when transferring the corresponding equation system to the coarse grid. This transfer has to be done in such a way that the coarse grid operator $A^{\mathbf{H}}(\mathbf{x}^{\mathbf{H}})$ of the coarse grid equation system

$$
A^{\mathbf{H}}(\mathbf{x}^{\mathbf{H}}) = \mathbf{b}^{\mathbf{H}} \tag{4.80}
$$

with new cell size $\mathbf{H} = (H_{x_1}, H_{x_2})^{\top}$ is still decomposable into

$$
A^{\mathbf{H}}(\mathbf{x}^{\mathbf{H}}) = B^{\mathbf{H}}(\mathbf{x}^{\mathbf{H}})\,\mathbf{x}^{\mathbf{H}} + \mathbf{c}^{\mathbf{H}}(\mathbf{x}^{\mathbf{H}}). \tag{4.81}
$$

Then, the same nonlinear solvers as for the single grid can be used. Apart from this, the construction of suitable unidirectional schemes is straightforward:

Intergrid Transfer Operator. As intergrid transfer operators the same restriction and prolongation techniques as in the linear case can be used: *full-weighting* or *area-based averaging* for restriction and *constant/linear* or *area-based interpolation* for prolongation.

Coarse Grid Equation System. However, in contrast to the linear case, only two different strategies are available to construct the coarse grid equation systems:

- Problem Based Coarse Grid Approximation (PCA)

- Discretisation Coarse Grid Approximation (DCA)

The Galerkin coarse grid approximation (GCA), that was introduced as the most accurate but also by far the most expensive coarse grid approximation, is not used in the context of nonlinear problems (at least not to restrict nonlinear equation systems). This has the following reason: Since $A^{\mathbf{H}}(\mathbf{x}^{\mathbf{H}})$ is a nonlinear operator, the construction via

$$A^{\mathbf{H}}(\mathbf{x}^{\mathbf{H}}) = R^{\mathbf{h}\rightarrow\mathbf{H}} A^{\mathbf{H}}(\mathbf{x}^{\mathbf{H}}) P^{\mathbf{H}\rightarrow\mathbf{h}} \tag{4.82}$$

cannot be represented any longer by a simple (though costly) multiplication of three matrices. In fact, in most cases this expression cannot be evaluated efficiently.

Concrete Example. Let us now show by the concrete example of our prototype F (method of Bruhn *et al.*), how such an unidirectional multigrid scheme can be derived for the nonlinear case. To this end, a suitable discretisation coarse grid approximation (DCA) approach is investigated. This is done in four steps.

I) *Basic Solver.* Firstly, we have to chose a basic iterative solver. Since our prototype F is based on isotropic regularisation, we propose to use the *Gauß-Seidel method with frozen coefficients and coupled point relaxation* (CPR) (cf. linear case). The corresponding iteration instructions have already been discussed. They are given by (4.74) - (4.79). The same solver can also be used for our prototypes D and G (isotropic regularisation as well). Only in the case of prototype E (anisotropic regularisation), a frozen coefficient variant of the Gauß-Seidel method with alternating line relaxation (ALR) may give better results.

II) *Coarse Grid Cell Size.* Secondly, we have to define the new coarse grid cell size **H**. As in the linear case, we favour a non-dyadic approach with

$$H_{x_1} := h_{x_1} \frac{N_{x_1}^{\mathbf{h}}}{N_{x_1}^{\mathbf{H}}} \qquad\qquad H_{x_2} := h_{x_2} \frac{N_{x_2}^{\mathbf{h}}}{N_{x_2}^{\mathbf{H}}} \tag{4.83}$$

where $N_{x_1}^{H} = \lceil N_{x_1}^{\mathbf{h}}/2 \rceil$ and $N_{x_2}^{\mathbf{H}} = \lceil N_{x_2}^{\mathbf{h}}/2 \rceil$. Please note that for our spatiotemporal prototype G (method of Papenberg *et al.*), the coarse grid cell size in temporal direction should be adapted in the same way.

III) *Intergrid Transfer Operators.* Thirdly, we have to define the interpolation and restriction operator. Again we choose *area based interpolation* and *area based averaging* (cf. linear case).

IV) *Coarse Grid Equation System.* Finally, we have to derive a suitable coarse grid representation for our fine grid equation system. Following the discretisation coarse grid approximation approach (DCA), this requires to rediscretise the discrete Euler–Lagrange equations. This yields the coarse grid equation system

$$
\left[\Psi'^{\mathbf{H}}_{\mathrm{D}}\right]_{i,j}\left[J^{\mathbf{H}}_{11}\right]_{i,j}\left[u^{\mathbf{H}}_{1}\right]_{i,j} + \left[\Psi'^{\mathbf{H}}_{\mathrm{D}}\right]_{i,j}\left[J^{\mathbf{H}}_{12}\right]_{i,j}\left[u^{\mathbf{H}}_{2}\right]_{i,j} + \left[\Psi'^{\mathbf{H}}_{\mathrm{D}}\right]_{i,j}\left[J^{\mathbf{H}}_{13}\right]_{i,j}
$$

$$
-\alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)} \frac{\left(\left[\Psi'^{\mathbf{H}}_{\mathrm{S}}\right]_{\tilde{i},\tilde{j}} + \left[\Psi'^{\mathbf{H}}_{\mathrm{S}}\right]_{i,j}\right)}{2} \frac{\left(\left[u^{\mathbf{H}}_{1}\right]_{\tilde{i},\tilde{j}} - \left[u^{\mathbf{H}}_{1}\right]_{i,j}\right)}{h_l^2} = 0
$$

$$(4.84)$$

$$
\left[\Psi'^{\mathbf{H}}_{\mathrm{D}}\right]_{i,j}\left[J^{\mathbf{H}}_{12}\right]_{i,j}\left[u^{\mathbf{H}}_{1}\right]_{i,j} + \left[\Psi'^{\mathbf{H}}_{\mathrm{D}}\right]_{i,j}\left[J^{\mathbf{H}}_{22}\right]_{i,j}\left[u^{\mathbf{H}}_{2}\right]_{i,j} + \left[\Psi'^{\mathbf{H}}_{\mathrm{D}}\right]_{i,j}\left[J^{\mathbf{H}}_{13}\right]_{i,j}
$$

$$
-\alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)} \frac{\left(\left[\Psi'^{\mathbf{H}}_{\mathrm{S}}\right]_{\tilde{i},\tilde{j}} + \left[\Psi'^{\mathbf{H}}_{\mathrm{S}}\right]_{i,j}\right)}{2} \frac{\left(\left[u^{\mathbf{H}}_{2}\right]_{\tilde{i},\tilde{j}} - \left[u^{\mathbf{H}}_{2}\right]_{i,j}\right)}{h_l^2} = 0
$$

$$(4.85)$$

for $i = 1, ..., N^{\mathbf{H}}_{x_1}$ and $j = 1, ..., N^{\mathbf{H}}_{x_2}$. As in the linear case, the entries of the coarse grid motion tensor $\left[J^{\mathbf{H}}\right]_{i,j}$ are obtained via channelwise restriction:

$$
\left[J^{\mathbf{H}}_{mn}\right]_{i,j} = R^{\mathbf{h}\to\mathbf{H}}\left[J^{\mathbf{h}}_{mn}\right]_{i,j} \qquad (m, n = 1, 2, 3). \tag{4.86}
$$

The entries of the diffusion tensors, however, depend on the $\mathbf{u}^{\mathbf{h}}_1$ and $\mathbf{u}^{\mathbf{h}}_2$ and are thus *nonlinear*. This in turn means that they cannot be restricted. Instead they have to be adapted to the new coarse grid cell size $\mathbf{H}$. For our prototype F this yields the nonlinear expressions

$$
\left[\Psi'^{\mathbf{H}}_{\mathrm{S}}\right]_{i,j} = \frac{1}{2\sqrt{|D^{2,\mathbf{H}}(\left[u^{\mathbf{H}}_{1}\right]_{i,j})|^2 + |D^{2,\mathbf{H}}(\left[u^{\mathbf{H}}_{2}\right]_{i,j})|^2 + \epsilon_{\mathrm{S}}^2}} , \tag{4.87}
$$

where $|D^{2,\mathbf{H}}(\left[z^{\mathbf{H}}\right]_{i,j})|$ denotes the gradient magnitude operator as defined in Table 3.1. Apart from the diffusion tensors also the nonlinear expressions from the nonquadratic data term have to be adapted to the coarse grid cell size. Analogously, they become

$$
\left[\Psi'^{\mathbf{H}}_{\mathrm{D}}\right]_{i,j} = \frac{1}{2\sqrt{(\left[u^{\mathbf{H}}_{1}\right]_{i,j}, \left[u^{\mathbf{H}}_{2}\right]_{i,j}, 1)^\top (J^{\mathbf{H}})_{i,j}(\left[u^{\mathbf{H}}_{1}\right]_{i,j}, \left[u^{\mathbf{H}}_{2}\right]_{i,j}, 1) + \epsilon_{\mathrm{D}}^2}} , \tag{4.88}
$$

where $\left[J^{\mathbf{H}}\right]_{i,j}$ is the restricted motion tensor from (4.86).

This example shows that our systematic framework for the design of efficient hierarchical schemes based on the motion and diffusion tensor notation can also be applied in the context of nonlinear unidirectional multigrid methods. If it is combined with a coarse grid approximation (DCA) approach, its construction rules become rather simple: While all *linear expressions* such as the motion tensor have to be *rediscretised* (they only depend on the input data), all *nonlinear expressions* are *adapted* to the new coarse grid cell size (they depend on the solution).
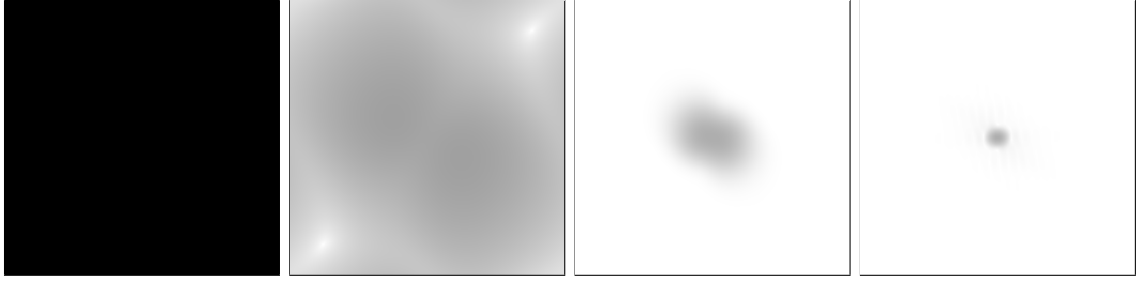
Fig. 4.10: Logarithmic Fourier spectra of the numerical error in the u-component of the optic flow for our prototype F. Development for an increasing number of standard Gauß-Seidel iterations. Lower frequencies are located in the centre of the image, higher frequencies at the boundaries. **(a) Outer Left:** Initialisation with equally distributed error frequencies. **(b) Centre Left:** 1 iteration. **(c) Centre Right:** 10 iterations. **(d) Outer Right:** 100 iterations.

4.2.4 Bidirectional Multigrid Methods

In the linear case we have identified the weak attenuation of low error frequencies as the actual reason for the poor performance of iterative splitting-based methods. As one can see from Figure 4.10 this also holds for nonlinear techniques that are derived from the Lagged-Diffusivity method. Hereby, apart from the original problem – the local coupling via a small neighbourhood – also the problem of the slow convergence of the fixed point iteration is present.

Basic Idea - The Nonlinear Bidirectional Two-Grid Cycle. For overcoming these problems so-called *full approximation schemes* (FAS) [Bra77] proved to be very useful. These bidirectional multigrid schemes for the nonlinear case rely on a similar error correction strategy than their linear counterparts. However, due the nonlinearity of the operator $A^{\mathbf{h}}(\tilde{\mathbf{x}}^{\mathbf{h}})$, their construction is slightly more difficult. Let us now explain by the example of a basic nonlinear bidirectional two-grid cycle how the FAS concept works.

I) *Presmoothing Relaxation.* As in the linear case we start by performing ν_1 iterations with a nonlinear iterative basic solver (e.g. a frozen coefficient method) in order to reduce the high frequency components of the error (see Figure 4.11). If we denote

Fig. 4.11: Logarithmic Fourier spectra of the numerical error for the Gauß-Seidel method with frozen coefficients and coupled point relaxation (prototype F). **(a) Left:** Before the presmoothing relaxation step. **(b) Right:** After the presmoothing relaxation (2 iterations).

the correct solution by $\mathbf{x^h}$ and the result after these ν_1 iterations by $\tilde{\mathbf{x}}^{\mathbf{h}}$, this error can be computed as follows

$$\mathbf{e^h} = \mathbf{x^h} - \tilde{\mathbf{x}}^{\mathbf{h}}. \tag{4.89}$$

II) *Coarse Grid Transfer.* Evidently, we are interested in computing $\mathbf{e^h}$ in order to correct our approximation. However, since $A^{\mathbf{h}}(\mathbf{x^h})$ is a nonlinear operator, the way of deriving a suitable coarse grid correction step is now completely different from the linear case. In fact, there is only an *implicit* relation between the error $\mathbf{e^h}$ and the residual $\mathbf{r^h} = \mathbf{f^h} - A^{\mathbf{h}}(\tilde{\mathbf{x}}^{\mathbf{h}})$. This relation is given by

$$A^{\mathbf{h}}(\tilde{\mathbf{x}}^{\mathbf{h}} + \mathbf{e^h}) - A^{\mathbf{h}}(\tilde{\mathbf{x}}^{\mathbf{h}}) = A^{\mathbf{h}}(\mathbf{x^h}) - A^{\mathbf{h}}(\tilde{\mathbf{x}}^{\mathbf{h}}) = \mathbf{f^h} - A^{\mathbf{h}}(\tilde{\mathbf{x}}^{\mathbf{h}}) = \mathbf{r^h}. \tag{4.90}$$

In order to compute the desired correction, we make use of this relation and transfer the following nonlinear equation system to the coarse grid:

$$A^{\mathbf{h}}(\underbrace{\tilde{\mathbf{x}}^{\mathbf{h}} + \mathbf{e^h}}_{\mathbf{x^h}}) = \underbrace{\mathbf{r^h} + A^{\mathbf{h}}(\tilde{\mathbf{x}}^{\mathbf{h}})}_{\tilde{\mathbf{f}}^{\mathbf{h}}} \quad \to \quad A^{\mathbf{H}}(\underbrace{\boxed{\tilde{\mathbf{x}}^{\mathbf{H}}} + \mathbf{e^H}}_{\mathbf{x^H}}) = \underbrace{\mathbf{r^H} + \boxed{A^{\mathbf{H}}(\tilde{\mathbf{x}}^{\mathbf{H}})}}_{\tilde{\mathbf{f}}^{\mathbf{H}}} . \tag{4.91}$$

In order to visualise the difference to the linear case the additional terms are marked by frames. As one can see, one cannot compute the error directly. However, one can determine $\tilde{\mathbf{x}}^{\mathbf{h}} + \mathbf{e^h}$ and via subtraction finally $\mathbf{e^h}$. If we reformulate the new right hand $\tilde{\mathbf{f}}^{\mathbf{H}}$ side as

$$
\begin{aligned}
\tilde{\mathbf{f}}^{\mathbf{H}} &= \mathbf{r^H} + A^{\mathbf{H}}(\tilde{\mathbf{x}}^{\mathbf{H}}) & (4.92) \\
&= R^{\mathbf{h}\to\mathbf{H}}\mathbf{r^h} + A^{\mathbf{H}}(\tilde{\mathbf{x}}^{\mathbf{H}}) \\
&= R^{\mathbf{h}\to\mathbf{H}}\big(\mathbf{f^h} - A^{\mathbf{h}}(\tilde{\mathbf{x}}^{\mathbf{h}})\big) + A^{\mathbf{H}}(\tilde{\mathbf{x}}^{\mathbf{H}}) \\
&= R^{\mathbf{h}\to\mathbf{H}}\mathbf{f^h} - R^{\mathbf{h}\to\mathbf{H}}A^{\mathbf{H}}(\tilde{\mathbf{x}}^{\mathbf{H}}) + A^{\mathbf{H}}(\tilde{\mathbf{x}}^{\mathbf{H}}) \\
&= \mathbf{f^H} + \big(A^{\mathbf{H}}(R^{\mathbf{h}\to\mathbf{H}}\tilde{\mathbf{x}}^{\mathbf{h}}) - R^{\mathbf{h}\to\mathbf{H}}A^{\mathbf{h}}(\tilde{\mathbf{x}}^{\mathbf{h}})\big) & (4.93)
\end{aligned}
$$

we can see that we are actually solving a modified variant of the original nonlinear equation system on the coarse grid. Please note that in general $\tilde{\mathbf{f}}^{\mathbf{H}} \neq \mathbf{f^H}$ since there is a significant difference between (i) evaluating the coarse grid operator for the restricted solution, i.e. computing $A^{\mathbf{H}}(R^{\mathbf{h}\to\mathbf{H}}\tilde{\mathbf{x}}^{\mathbf{h}})$ and (ii) restricting the result of the evaluated fine grid operator for the fine grid solution, i.e. computing $R^{\mathbf{h}\to\mathbf{H}}A^{\mathbf{h}}(\tilde{\mathbf{x}}^{\mathbf{h}})$.

III) *Coarse Grid Solution.* As in the linear case the remaining low frequency components after the presmoothing relaxation step reappear as higher frequencies in the error of our coarse grid problem. Therefore, they can now be attenuated efficiently by using the nonlinear iterative basic solver again.

IV) *Fine Grid Transfer.* Since solving the nonlinear residual equation system on the coarse grid gives us actually $\tilde{\mathbf{x}}^{\mathbf{H}} + \mathbf{e^H}$, we have to subtract $\tilde{\mathbf{x}}^{\mathbf{H}}$ from the computed

solution in order to obtain the true coarse grid error $\mathbf{e^H}$. This solution has then to be transferred to the fine grid

$$\mathbf{e^H} \rightarrow \tilde{\mathbf{e}}^\mathbf{h}. \tag{4.94}$$

Once again, the interpolated coarse grid solution is denoted by $\tilde{\mathbf{e}}^\mathbf{h}$, since it is only an approximation to the desired correction $\mathbf{e^h}$.

V) *Coarse Grid Correction.* After having transferred the coarse grid solution to the fine step, it can be used to correct our approximative result from the presmoothing relaxation step:

$$\tilde{\mathbf{x}}^\mathbf{h}_{\text{new}} = \tilde{\mathbf{x}}^\mathbf{h} + \tilde{\mathbf{e}}^\mathbf{h}. \tag{4.95}$$

As we can see from the corresponding Fourier spectra shown in Figure 4.12 this results in the desired attenuation of low error frequencies. The interpolation, however, introduces some new high frequencies.
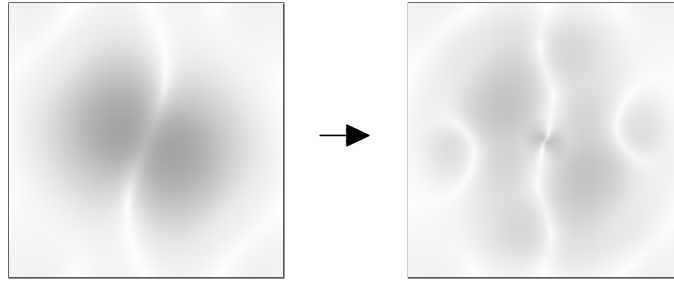
Fig. 4.12: Logarithmic Fourier spectra of the numerical error for the Gauß-Seidel method with frozen coefficients and coupled point relaxation (prototype F). **(a) Left:** Before the coarse grid correction step. **(b) Right:** After the coarse grid correction step.

VI) *Postsmoothing Relaxation.* In order to remove this high error frequencies, finally ν_2 iterations of the nonlinear iterative basic solver are applied (see Figure 4.13).
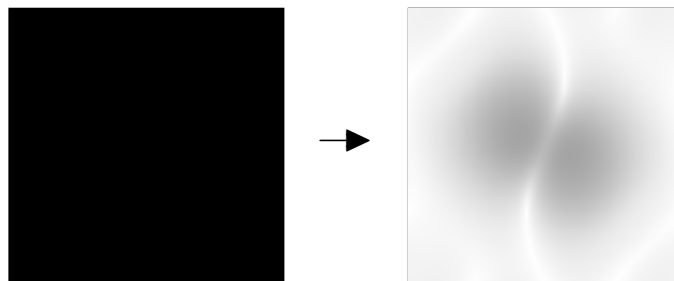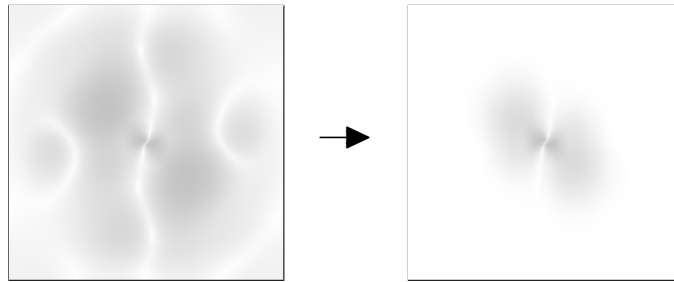

Fig. 4.13: Logarithmic Fourier spectra of the numerical error for the Gauß-Seidel method with frozen coefficients and coupled point relaxation (prototype F). **(a) Left:** Before the postsmoothing relaxation step. **(b) Right:** After the postsmoothing relaxation step (2 iterations).

As one can see, the basic idea of full approximation schemes is very similar to the one of bidirectional multigrid methods for the linear case. Let us now discuss how the necessary intergrid transfer operators have to be chosen and how one can obtain the required coarse grid equation system.
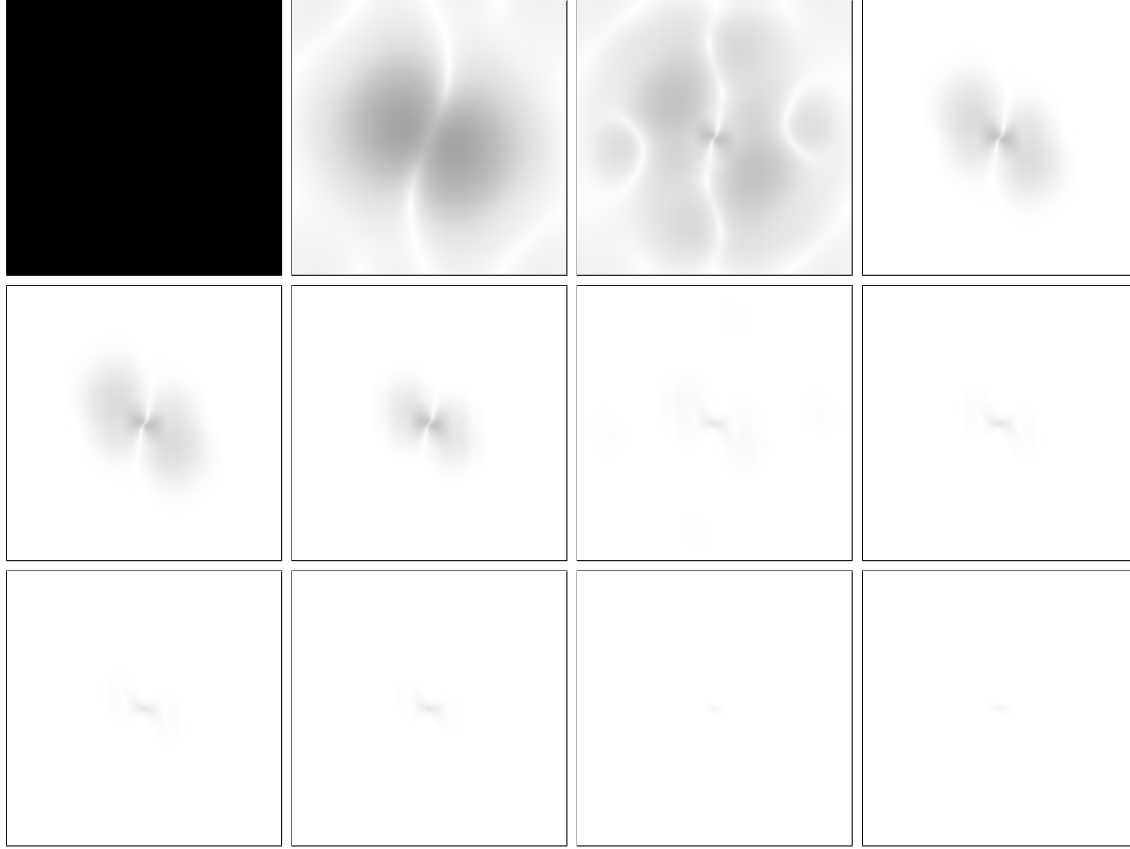
Fig. 4.14: Logarithmic Fourier spectra of the numerical error in the u-component of the optic flow for our prototype F. Error reduction over three V-cycles. **From Top to Bottom:** First cycle, second cycle and third cycle. **From Left to Right:** Initialisation, presmoothing relaxation step, coarse grid correction step, postsmoothing relaxation step.

Intergrid Transfer Operators. In order to allow for the construction of efficient multigrid schemes, the intergrid transfer operators have to fulfil the same accuracy condition as in the linear case (4.52). Moreover, they should be chosen in such a way that they preserve the positive definiteness of the coarse grid motion tensors – this allows the application of the same nonlinear iterative basic solver on all multigrid levels. Reasonable choices are the *full-weighting* or *area-based averaging* operator for restriction and *constant/linear* or *area-based interpolation* operator for prolongation.

Coarse Grid Equation System. As already explained in the unidirectional case, only two strategies are used in the context of nonlinear problems to derive the coarse grid system:

- Problem Based Coarse Grid Approximation (PCA)

- Discretisation Coarse Grid Approximation (DCA)

Since the approximation quality of the problem based coarse grid approximation is often so poor that the resulting bidirectional multigrid methods do not converge, one should only rely on approaches based on the discretisation coarse grid approximation (DCA).

Hierarchical Application. So far we have only discussed the basic two-grid cycle. However, analogously to the linear case, this cycle can be extended to more advanced grid traversal strategies that have better error reduction properties such as V- and W-cycles (cf. Figure 4.7). As in the linear case, two to three cycles are already sufficient to achieve a significant attenuation of all frequencies and thus a decay of the Euclidean norm of the error of about *four orders of magnitude* (see Figure 4.14). However, V- and W-cycles alone are rarely used in the context of nonlinear problems. They are often embedded within a *full multigrid scheme* (cf. Figure 4.9). This has the following reason: Unlike in our case where the solution is unique, often nonlinear problems have *multiple solutions*. In order to converge to the "right"'solution, in particular with respect to Newton-like techniques, it is important to be sufficiently close to the right basin of attraction when starting the iterations. This in turn can be achieved by the hierarchical initialisation that used by full multigrid schemes.

Concrete Example. Let us now discuss how such a DCA-based full approximation scheme can be derived for our prototype F (method of Bruhn *et al.*). One should note apart from the control-theory-based optic flow approach in [BIK02], no attempts have been made so far to use nonlinear bidirectional multigrid methods in the context of variational optic flow computation. In order to set up the full approximation scheme, the following four decisions have to be made.

I) *Basic Solver*. Firstly, we have to decide for a basic iterative solver. As in the unidirectional case, we propose the use of a *Gauß-Seidel method with frozen coefficients and coupled point relaxation* (CPR) for this purpose.

II) *Intergrid Transfer Operators*. Secondly, a suitable pair of intergrid transfer operators has to be defined. While the restriction operator is required to transfer the residual *and* the approximative solution to the coarser grid as well as for the construction of the coarse grid motion tensors, the interpolation operator is needed to interpolate the obtained coarse grid correction. As we have shown in the case of linear bidirectional methods, *area based interpolation* and *area based averaging* are a suitable choice [BWF$^+$05, BWKS06]: Firstly, they preserve the positive definiteness of the coarse grid motion tensors and therefore allow the application of frozen coefficient methods, and secondly and equally important, they fulfil the accuracy condition in (4.52).

III) *Coarse Grid Cell Size*. Thirdly, a new coarse grid cell size **H** has to be chosen. Also this time, we propose a non-dyadic approach with

$$H_{x_1} := h_{x_1} \frac{N_{x_1}^{\mathbf{h}}}{N_{x_1}^{\mathbf{H}}} \qquad\qquad H_{x_2} := h_{x_2} \frac{N_{x_2}^{\mathbf{h}}}{N_{x_2}^{\mathbf{H}}} \qquad\qquad (4.96)$$

with $N_{x_1}^H = \lceil N_{x_1}^{\mathbf{h}}/2 \rceil$ and $N_{x_2}^{\mathbf{H}} = \lceil N_{x_2}^{\mathbf{h}}/2 \rceil$. In the case of our spatiotemporal prototype G one should not forget to treat the temporal axis in the same way, i.e. to define

$$H_{x_3} := h_{x_3} \frac{N_{x_3}^{\mathbf{h}}}{N_{x_3}^{\mathbf{H}}} \qquad\qquad (4.97)$$

with $N_{x_3}^H = \lceil N_{x_3}^{\mathbf{h}}/2 \rceil$.

IV) *Residual Coarse Grid Equation System.* Finally, we have to set up the *nonlinear residual equation system* for the coarse grid. Once again, we follow the discretisation coarse grid approximation approach (DCA) and rediscretise all entries of the discrete Euler–Lagrange equations that belong to the nonlinear operator $A^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})$. This yields the coarse grid equation system

$$
\left[\Psi_{\mathrm{D}}'^{\mathbf{H}}\right]_{i,j}\left[J_{11}^{\mathbf{H}}\right]_{i,j}\left[u_1^{\mathbf{H}}\right]_{i,j} + \left[\Psi_{\mathrm{D}}'^{\mathbf{H}}\right]_{i,j}\left[J_{12}^{\mathbf{H}}\right]_{i,j}\left[u_2^{\mathbf{H}}\right]_{i,j} + \left[\Psi_{\mathrm{D}}'^{\mathbf{H}}\right]_{i,j}\left[J_{13}^{\mathbf{H}}\right]_{i,j}
$$

$$
-\alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)} \frac{\left(\left[\Psi_{\mathrm{S}}'^{\mathbf{H}}\right]_{\tilde{i},\tilde{j}} + \left[\Psi_{\mathrm{S}}'^{\mathbf{H}}\right]_{i,j}\right)\left(\left[u_1^{\mathbf{H}}\right]_{\tilde{i},\tilde{j}} - \left[u_1^{\mathbf{H}}\right]_{i,j}\right)}{2} = \left[\tilde{f}_1^{\mathbf{H}}\right]_{i,j}
$$

$$(4.98)$$

$$
\left[\Psi_{\mathrm{D}}'^{\mathbf{H}}\right]_{i,j}\left[J_{12}^{\mathbf{H}}\right]_{i,j}\left[u_1^{\mathbf{H}}\right]_{i,j} + \left[\Psi_{\mathrm{D}}'^{\mathbf{H}}\right]_{i,j}\left[J_{22}^{\mathbf{H}}\right]_{i,j}\left[u_2^{\mathbf{H}}\right]_{i,j} + \left[\Psi_{\mathrm{D}}'^{\mathbf{H}}\right]_{i,j}\left[J_{13}^{\mathbf{H}}\right]_{i,j}
$$

$$
-\alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)} \frac{\left(\left[\Psi_{\mathrm{S}}'^{\mathbf{H}}\right]_{\tilde{i},\tilde{j}} + \left[\Psi_{\mathrm{S}}'^{\mathbf{H}}\right]_{i,j}\right)\left(\left[u_2^{\mathbf{H}}\right]_{\tilde{i},\tilde{j}} - \left[u_2^{\mathbf{H}}\right]_{i,j}\right)}{2} = \left[\tilde{f}_2^{\mathbf{H}}\right]_{i,j}
$$

$$(4.99)$$

for $i = 1, ..., N_{x_1}^{\mathbf{H}}$ and $j = 1, ..., N_{x_2}^{\mathbf{H}}$, where $\left[u_1^{\mathbf{H}}\right]_{i,j}$ and $\left[u_2^{\mathbf{H}}\right]_{i,j}$ are the new unknowns (coarse grid solution) and $\left[\tilde{f}_1^{\mathbf{H}}\right]_{i,j}$ and $\left[\tilde{f}_2^{\mathbf{H}}\right]_{i,j}$ give the modified right hand side. As in the unidirectional case, the entries of the coarse grid motion tensors $\left[J^{\mathbf{H}}\right]_{i,j}$ are obtained via channelwise restriction, while the nonlinear expressions $\left[\Psi_{\mathrm{D}}'^{\mathbf{H}}\right]_{i,j}$ and $\left[\Psi_{\mathrm{S}}'^{\mathbf{H}}\right]_{i,j}$ are adapted with respect to the new coarse grid cell size $\mathbf{H}$ (the latter one is a consequence from adapting the diffusion tensor).

Evidently, our coarse grid construction rules from the unidirectional case can also be applied in the context of bidirectional multigrid methods (FAS schemes). This in turn means that by combining the diffusion and motion tensor notation with a suitable discretisation coarse grid approximation approach (DCA), we have provided a complete framework for the design of efficient multigrid schemes for variational optic flow methods for small displacements. One should note that this framework is not restricted to the seven prototypes that have been selected. It can be applied to the whole class of variational techniques that can be designed in accordance with our theoretical guidelines in Chapter 2.

4.3   Numerical Prototypes

In the previous two sections we have discussed a variety of numerical schemes for the solution of the linear and nonlinear equations systems associated to our prototypes A-G. However, as for the variational models, only a few prototypes can actually be tested in our numerical experiments. Therefore, we selected six numerical techniques for each of the systems of equations A-G: One bidirectional and one unidirectional multigrid method (both based on our multigrid framework), one basic and one advanced non-hierarchical solver, and two techniques that make use of the briefly discussed elliptic minimisation strategy (these techniques compute the steady state of the steepest descent equations; cf. 2.3.1.1). Let us now discuss the different implementations in detail.

Tab. 4.1: Implemented bidirectional multigrid schemes for the different variational models. MG = multigrid. FMG = full multigrid. W = W-cycle. Cyc = multigrid cycles per level. GS = Gauß–Seidel. CPR = coupled point relaxation. ALR = alternating line relaxation. Pre/Post = pre- and postsmoothing relaxation iterations. L = linear. NL = nonlinear. SD = small displacements

Case	Model	MG Solver	Cyc	Basic Solver	Pre/Post
L	A - Homogeneous	FMG-W	1	GS-CPR	1-1
	B - Image-Driven Isotropic	FMG-W	2	GS-CPR	2-2
	C - Image-Driven Anisotropic	FMG-W	4	GS-ALR	1-1 -
NL I	D - Flow-Driven Isotropic	FAS-FMG-W	2	GS-CPR	2-2
	E - Flow-Driven Anisotropic	FAS-FMG-W	4	GS-ALR	1-1
NL II	F - Bruhn *et al.* 2-D, SD	FAS-FMG-W	2	GS-CPR	2-2
	G - Papenberg *et al.* 3-D, SD	FAS-FMG-W	2	GS-CPR	2-2

Multigrid Solvers. As representatives for the class of bidirectional multigrid methods we developed several full multigrid schemes (see Table 4.1). Thereby we used two different types of basic iterative solvers: While in the cases of homogeneous and isotropic regularisation, a Gauß–Seidel technique with coupled point relaxation (CPR) was sufficient, the anisotropy of the neighbourhood coupling required the use of a Gauß–Seidel method with alternating line relaxation (ALR). For the nonlinear variants of the Gauß–Seidel solvers we used the strategy of frozen coefficients. Direct nonlinear Gauß–Seidel Newton methods [BHM00] have not been considered as basic solver. Preliminary experiments using this kind of methods have shown a similar performance in terms of error reduction, however, at the expense of significantly increased computational costs.

As one can see from Table 4.1 an increasing anisotropy of the diffusion tensor (homogeneous → isotropic → anisotropic) requires more multigrid cycles at each level of the full multigrid implementation. This can be explained by the deterioration of the approximation quality of the coarse grid operator if the entries of the diffusion tensor vary strongly.

In order to allow for a complete multigrid hierarchy and the application to image sequences of any size, we considered the use of non-dyadic intergrid transfer operators in all approaches. As we proposed before, they were realised by area-based interpolation and area-based averaging. Coarser versions of the linear and nonlinear operators were created by the discretisation coarse grid approximation approach (DCA) [Wes92]. To this end, we restricted all linear expressions such as the motion tensors and the linear diffusion tensors and adapted the grid size for the nonlinear ones.

In addition to our full multigrid implementations we also developed unidirectional multigrid schemes for all our prototypes (cascadic multigrid methods). In order to allow a fair comparison to their bidirectional counterparts, we used the same basic iterative solvers (linear and nonlinear Gauß-Seidel variants with coupled point (CPR) and alternating line relaxation (ALR)). Moreover, we used the same intergrid transfer operators and the same coarse grid approximation strategy (DCA).

Tab. 4.2: Overview over all implemented solvers. NL = variant for the nonlinear case. FC = frozen coefficients. ILD = inexact lagged diffusivity (> 1 iterations between nonlinear updates).

Case	Solver Type	Selected Solver	NL
Elliptic	Basic Non-Hierarchical	Gausß-Seidel (CPR/ALR)	FC
	Advanced Non-Hierarchical	Successive Overrelaxation (SOR)	ILD
	Unidirectional Multigrid	Cascadic Gauß-Seidel (CPR/ALR)	FC
	Bidirectional Multigrid	Full Multigrid (see Table 4.1)	FC
Parabolic	Explicit Scheme	Stabilised Explicit Scheme	-
	Semi-Implicit Scheme	Semi-Implicit Scheme + SOR	-

Non-Hierarchical Iterative Solvers. As contestants for the class of non-hierarchical iterative solvers we selected the following two prototypes: the previously used *Gauß-Seidel variants* as basic non-hierarchical solvers and the *successive overrelaxation method* (SOR) as advanced non-hierarchical technique. In the latter case we realised the nonlinear variant by means of an inexact lagged diffusivity method.

Parabolic Solvers. Finally, as numerical prototypes for the class of parabolic solvers – such methods compute the steady state of the steepest descent equations – we implemented a *stabilised explicit* [WS01b] and a *semi-implicit* approach [WHS$^+$01]. While the first method is a straightforward extension of a standard explicit scheme (e.g. a gradient descent method) that allows for larger time step sizes τ, the second approach is unconditionally stable but requires the solution of a linear equation system at each step (e.g. by using the SOR method). For a comparison of different numerical schemes for the parabolic case the interested reader is referred to [WHS$^+$01]. An overview of all six techniques that we have implemented for our comparison is given in Table 4.2.

4.4   Numerical Experiments

Having discussed all numerical prototypes, let us now evaluate the efficiency of our multigrid implementations. Thereby all computations are carried out on a standard desktop PC with a 3.06 GHz Intel Pentium 4 CPU executing C / C++ code.

4.4.1   Basic Variational Approaches

In our first experiment we compare the efficiency of different numerical schemes for the five basic prototypes A-E. For our evaluation we used a 160×120 real-world sequence (the *Dancing* sequence; see Figure 4.15), where a person dances in front of the camera. Before we applied the different numerical schemes we preprocessed the sequence by convolution with a Gaussian kernel of standard deviation $\sigma = 1$. The iterations were stopped when the relative error $e_{rel} := \|x - \tilde{x}_n\|_2 / \|x\|_2$ dropped below 10^{-2}, where x denotes the correct numerical solution and $\tilde{x}_n$ stands for the computed result after n iterations/cycles. The correct numerical solution was thereby computed in advance with a very high precision. In all cases a zero flow field served as initialisation.

Tab. 4.3: **Experiment I:** Performance benchmark for the linear case (prototypes A-C) on a standard desktop computer with 3.06 GHz Pentium 4 CPU. Run times refer to the computation of a single flow field from the 160×120 dancing sequence. FPS = frames per second.

Linear Case: Prototype A (homogeneous regularisation)
$\sigma = 1.0$, $\alpha = 1000$

Solver	Iterations	Time $[s]$	FPS $[s^{-1}]$	Speedup
Modified Explicit Scheme ($\tau = 0.25$)	4425	3.509	0.285	1
Gauß-Seidel (CPR)	2193	1.152	0.868	3
Semi-Implicit Scheme ($\tau = 1000$) + SOR	82/1	0.073	13.699	48
SOR ($\omega = 1.94$)	82	0.052	19.233	67
Cascadic Gauß-Seidel (CPR)	16	0.018	56.189	197
Full Multigrid	**1**	**0.016**	**62.790**	**220**

Linear Case: Prototype B (image-driven isotropic regularisation)
$\sigma = 1.0$, $\alpha = 1000$, $\epsilon_{\mathrm{S}} = 1.0$

Solver	Iterations	Time $[s]$	FPS $[s^{-1}]$	Speedup
Mod. Explicit Scheme ($\tau = 0.25$)	8896	12.048	0.083	1
Gauß-Seidel (CPR)	2856	2.793	0.358	4
Cascadic Gauß-Seidel (CPR)	215	0.085	3.508	42
Semi-Implicit Scheme ($\tau = 1000$) + SOR	89/1	0.124	8.065	97
SOR ($\omega = 1.95$)	91	0.103	9.709	117
Full Multigrid	**1**	**0.048**	**20.850**	**251**

Linear Case: Prototype C (image-driven anisotropic regularisation)
$\sigma = 1.0$, $\alpha = 1000$, $\epsilon_{\mathrm{S}} = 10^{-2}$

Solver	Iterations	Time $[s]$	FPS $[s^{-1}]$	Speedup
Mod. Explicit Scheme ($\tau = 0.1666$)	36433	47.087	0.021	1
Gauß-Seidel (ALR)	607	3.608	0.277	13
Cascadic Gauß-Seidel (ALR)	473	3.218	0.311	15
Semi-Implicit Scheme ($\tau = 1000$) + SOR	199/1	0.244	4.098	195
SOR ($\omega = 1.96$)	202	0.212	4.417	224
Full Multigrid	**1**	**0.171**	**5.882**	**275**

The Linear Case. Table 4.3 shows the excellent performance of our full multigrid schemes for the linear case (prototypes A-C): For the *first time* in the literature, *real-time frame rates* are achieved for variational optic flow methods with homogeneous and image-driven regularisation (up to 63 dense flow fields per second). Thereby speedups of two to three and one to two orders of magnitude are obtained compared to frequently used modified explicit schemes and widely spread basic iterative solvers, respectively. Also compared to their unidirectional counterparts our full multigrid implementations performed favourably: Except for the homogeneous case (prototype A) where the results are usually rather smooth and the hierarchical initialisation works almost perfectly, we achieved average speedups

Tab. 4.4: **Experiment I:** Performance benchmark for the linear case (prototypes D-E) on a standard desktop computer with 3.06 GHz Pentium 4 CPU. Run times refer to the computation of a single flow field from the 160×120 dancing sequence. FPS = frames per second.

Nonlinear Case I: Prototype D (flow-driven isotropic regularisation)
$\sigma = 1.0, \alpha = 10, \epsilon_S = 10^{-2}$

Solver	Iterations	Time $[s]$	FPS $[s^{-1}]$	Speedup
Mod. Explicit Scheme ($\tau = 0.0025$)	10633	30.492	0.033	1
Gauß-Seidel (CPR)	2679	6.911	0.145	4
Cascadic Gauß-Seidel (CPR)	371	0.853	1.173	36
Semi-Implicit Scheme ($\tau = 10000$) + SOR	17/5	0.206	4.854	148
Inexact Lagged SOR ($\omega = 1.95$)	17/5	0.174	5.748	174
FAS - Full Multigrid	**1**	**0.082**	**12.172**	**372**

Nonlinear Case I: Prototype E (flow-driven anisotropic regularisation)
$\sigma = 1.0, \alpha = 10, \epsilon_S = 10^{-2}$

Solver	Iterations	Time $[s]$	FPS $[s^{-1}]$	Speedup
Mod. Explicit Scheme ($\tau = 0.0025$)	9224	58.837	0.017	1
Gauß-Seidel (ALR)	591	12.508	0.080	5
Cascadic Gauß-Seidel (ALR)	138	3.816	0.262	15
Semi-Implicit Scheme ($\tau = 10000$) + SOR	35/5	0.595	1.681	99
Inexact Lagged SOR ($\omega = 1.90$)	35/5	0.509	1.964	116
FAS - Full Multigrid	**1**	**0.491**	**2.038**	**120**

of one order of magnitude. Only the SOR based numerical schemes with manually optimised overrelaxation parameter ω showed a more or less comparable performance (due to the low but realistic accuracy requirements of $e_{rel} = 10^{-2}$): While the SOR method is still outperformed by a factor three for our prototype A, it can reduce the gap to a difference of about 20% in the anisotropic case (prototype C). As explained in Section 4.3, this is directly related to the reduced approximation quality of the coarse grid operator within our full multigrid schemes for highly anisotropic motion tensors with varying coefficients. One should note that the more sophisticated Galerkin coarse-grid approximation strategy (GCA) may yield a faster convergence in terms of required cycles on each multigrid level, but is so expensive that it does not allow to reach real-time frame rates.

Apart from the good performance of our full multigrid implementations, this experiment also shows that it *does not give any advantage* to solve the obtained equation systems using a parabolic approach. In fact, the semi-implicit scheme yielded the best results when a very large time step size was used ($\tau = 1000$) and only one iteration of the SOR method was applied to solve the resulting linear equation system in each time step. This in turn means that *we actually solve an elliptic problem ($\tau \to \infty$)*, and this with a standard SOR solver. Thus it is not surprising that the obtained results of the semi-implicit scheme are basically identical with the ones of the ordinary SOR method (apart from slightly increased computational costs).
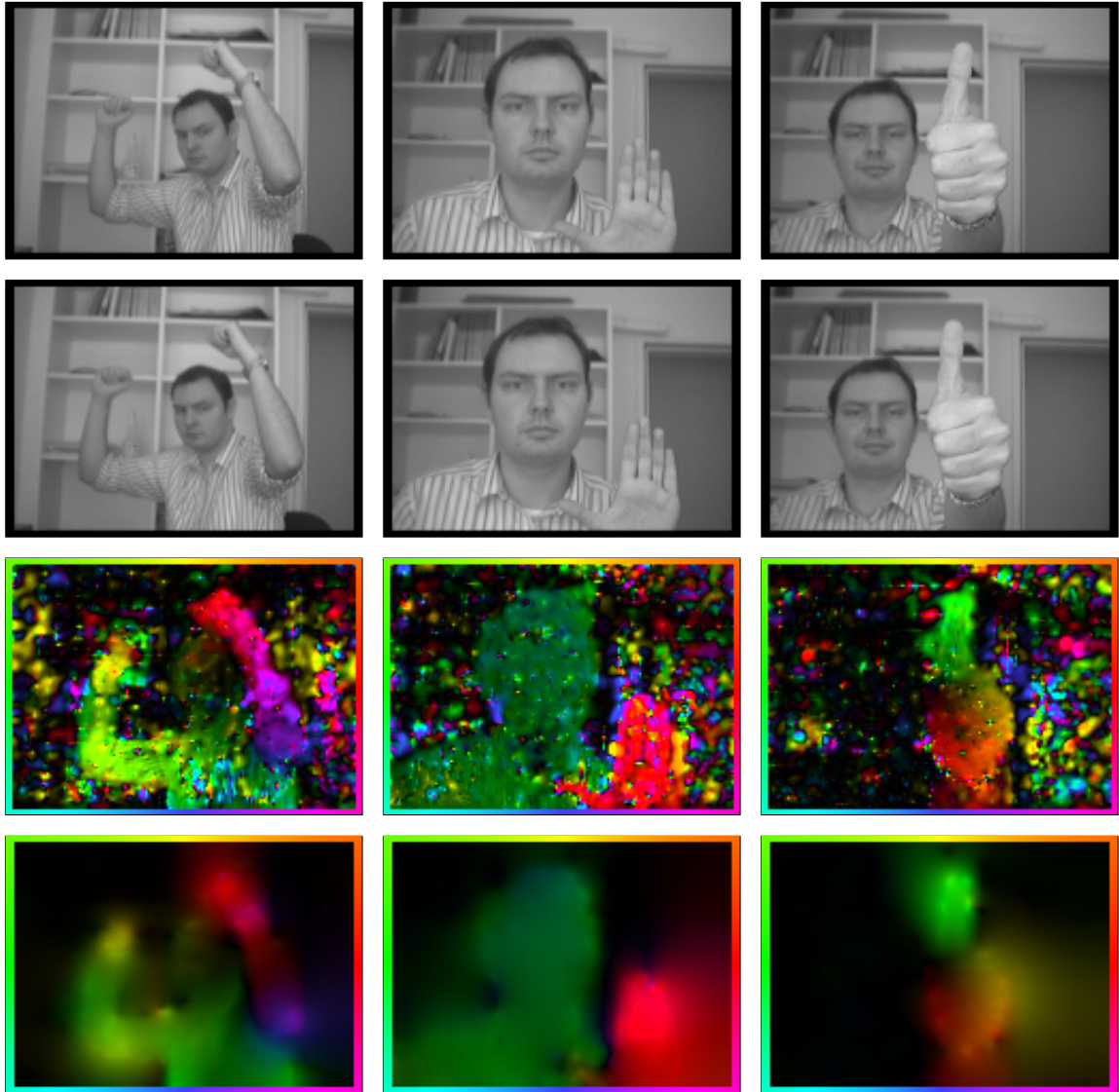
Fig. 4.15: **Experiment II:** Quality of our real-time implementations. **Left to Right:** dancing sequence, waving sequence, rotating thumb sequence. **Top to Bottom:** First frame, Second frame, No regularisation (normal flow), Prototype A (homogeneous regularisation).

The Nonlinear Case I. In the nonlinear case (prototypes D and E) our comparison shows a very similar tendency (see Table 4.4). Also for flow-driven methods *real-time* frame rates are achieved for the first time in the literature (up to 12 frames per second). Here, the speedup factors are in the range of two to three orders of magnitude compared to modified explicit schemes and basic iterative solvers and about one order of magnitude compared to unidirectional multigrid techniques. Also in this case the SOR method showed a good performance: While it was outperformed by a factor two in the isotropic case (prototype D) it proved to be almost as efficient as our full multigrid solver in the anisotropic case (prototype E). Regarding the comparison between elliptic and parabolic approaches, we can observe again that the parabolic strategy does not give any advantage: As before the semi-implicit scheme comes down to an *elliptic solver* when it achieves its best performance. Then, it approximates the inexact lagged diffusivity method for the SOR scheme.
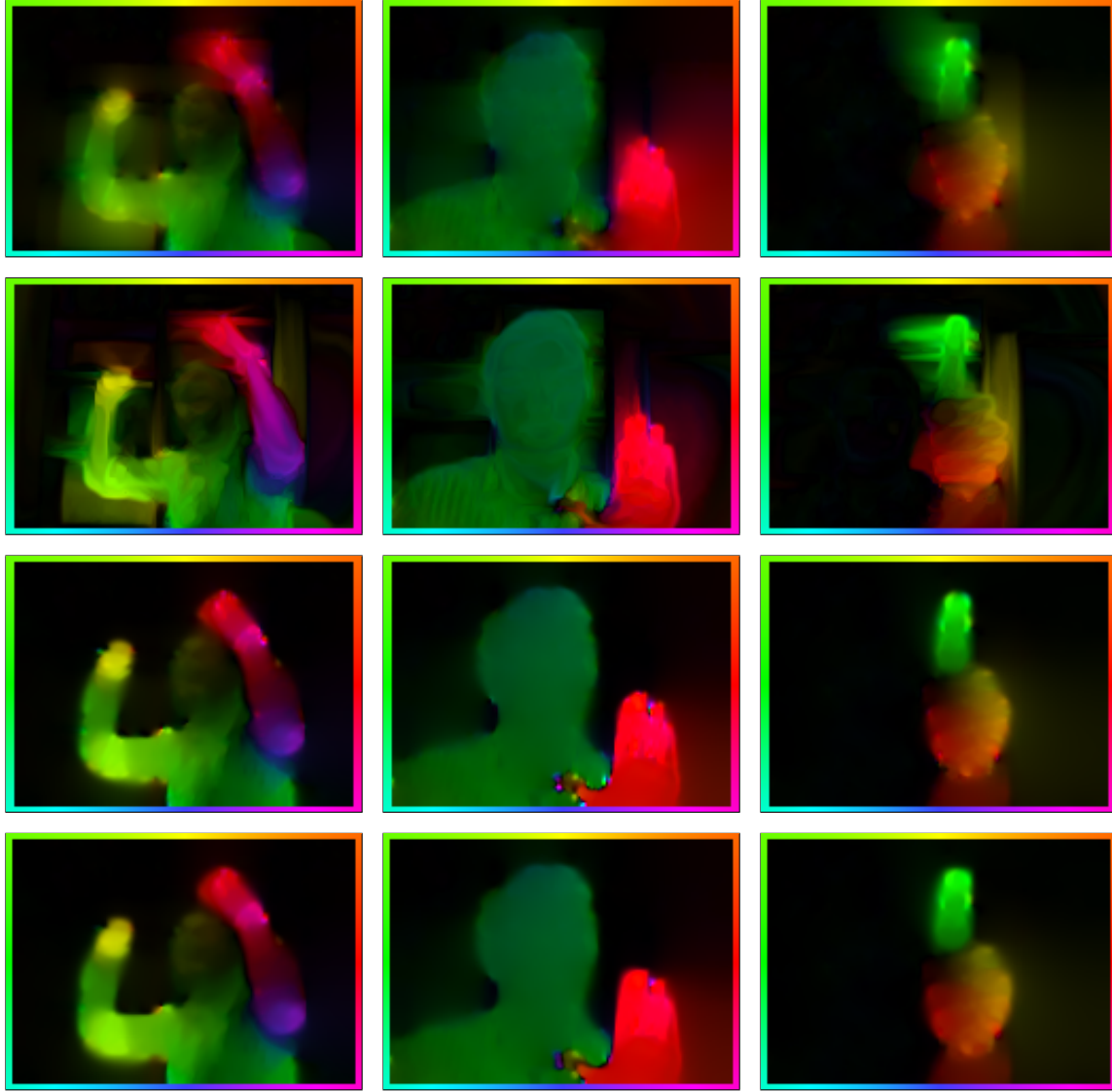
Fig. 4.16: **Experiment II:** Quality of our real-time implementations. *Left to right:* dancing sequence, waving sequence, rotating thumb sequence. *Top to bottom:* Prototype B (data-driven isotropic regularisation), Prototype C (data-driven anisotropic regularisation), Prototype D (flow-driven isotropic regularisation), Prototype E (flow-driven anisotropic regularisation).

Quality. In our second experiment we juxtapose the estimation quality of our proposed full multigrid implementations for the different regularisation strategies (prototypes A-E). To this end, we computed flow fields for three different real-world sequences: for the previously used *Dancing* sequence (complex motion), the *Waving* sequence (translations and discontinuities) and the *Rotating Thumb* sequence (rotation). The depicted flow fields in Figure 4.15 and Figure 4.16 make the quality of our real-time variational optic flow methods explicit: One can see that the image- and flow-driven prototypes B-E allow the preservation of discontinuities, that the anisotropic prototypes C and E give slightly better results than their isotropic counterparts B and D, and that the flow-driven methods E and F are able to overcome the problem of oversegmentation that lies in the nature of image-driven techniques for heavily textured scenes.

Tab. 4.5: **Experiment III:** Performance benchmark for the nonlinear case II (prototypes F-G) on a standard desktop computer with 3.06 GHz Pentium 4 CPU. Run times refer to the computation of a single flow field from the downsampled *Rheinhafen* sequence (size 160×120; see Figure 4.17). FPS = frames per second.

<div align="center">

Nonlinear Case II: Prototype F (Bruhn *et al.* 2-D, SD)
Robust data term with local integration + flow-driven isotropic regularisation
$\sigma = 0.0, \rho = 1.0, \alpha = 15, \epsilon_D = 10^{-1}, \epsilon_S = 10^{-3}$

</div>

Solver	Iterations	Time $[s]$	FPS $[s^{-1}]$	Speedup
Mod. Explicit Scheme ($\tau = 0.00025$)	81064	246.812	0.004	1
Gauß-Seidel (CPR)	3720	9.524	0.105	26
Cascadic Gauß-Seidel (CPR)	138	0.409	2.448	603
Semi-Implicit Scheme ($\tau = 10000$) + SOR	20/5	0.243	4.115	1016
Inexact Lagged SOR ($\omega = 1.96$)	20/5	0.189	5.291	1306
FAS - Full Multigrid	**1**	**0.087**	**11.473**	**2836**

<div align="center">

Nonlinear Case II: Prototype G (Papenberg *et al.* 3-D, SD) :
Robust data term with additional gradient constancy + flow-driven isotropic regularisation
$\sigma = 1.0, \alpha = 30, \gamma = 15, \epsilon_D = 10^{-1}, \epsilon_S = 10^{-3}$

</div>

Solver	Iterations	Time $[s]$	FPS $[s^{-1}]$	Speedup
Mod. Explicit Scheme ($\tau = 0.00025$)	184913	6408.660	0.001	1
Gauß-Seidel (CPR)	10809	231.883	0.026	28
Cascadic Gauß-Seidel (CPR)	595	14.807	0.405	432
Semi-Implicit Scheme ($\tau = 10000$) + SOR	23/5	2.083	2.880	3076
Inexact Lagged SOR ($\omega = 1.93$)	23/5	1.826	3.286	3510
FAS - Full Multigrid	**1**	**0.607**	**9.885**	**10558**

4.4.2   Advanced Variational Methods

In our third experiment we investigate the efficiency of our multigrid implementations for the more advanced variational optic flow prototypes F and G. As test sequence in this experiment served a downsampled variant (160×120) of the *Rheinhafen* sequence. As before, a relative error of $e_{rel} = 10^{-2}$ was used as stopping criterion.

The Nonlinear Case II. Let us now take a look at the obtained results in Table 4.5. The corresponding frame rates show clearly that even in the case of such highly advanced optic flow methods, real-time performance is still possible. Moreover, one can see that the speedups for the more advanced optic flow methods are even better than for the basic techniques with different types of regularisation. One should particularly note the large speedups for the spatiotemporal prototype G: With three to four orders of magnitude the modified explicit scheme (that needs one to two hundred thousand iterations) is outperformed more than significantly. The same applies to the basic iterative solvers, although the speedup is here "only" two orders of magnitude. Even compared to their unidirectional counterparts, the acceleration is still in the range of one order of magnitude. As
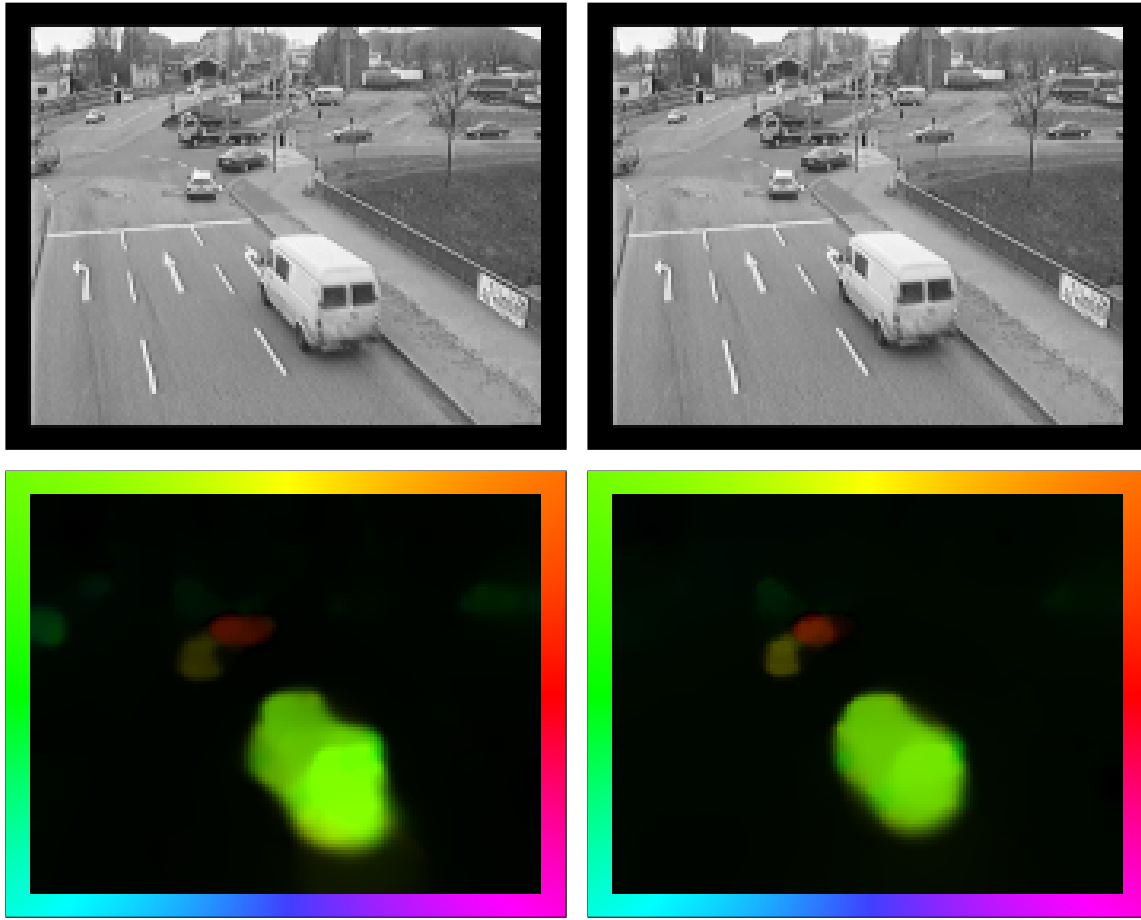
Fig. 4.17:   **(a) Top Left:** Frame 1130 of the *Rheinhafen* (resized to 160×120). **(b) Top Right:** Frame 1131. **(c) Bottom Left:** Flow field computed by our full multigrid implementation for the prototype F (Bruhn *et al.* 2-D, SD). Computing time: 87 milliseconds. **(d) Bottom Right:** Flow field computed by our full multigrid implementation for the prototype G (Papenberg *et al.* 3-D, SD). Computing time: 121 milliseconds.

in the previous benchmark for the prototypes A-E the SOR technique showed a relatively good performance. This time it is only outperformed by a factor two to three. The same holds for the semi-implicit scheme. As before, this parabolic strategy required a very large time step size to yield its optimal performance and thus came once more down to a solver for the elliptic strategy.

Quality. In Figure 4.17 the computed flow fields are depicted. Evidently they look fairly realistic: The motion of the van in the foreground as well as the motion of most other vehicles in the background is computed with good precision. Moreover, object boundaries within the flow field are rather sharp and allow for a simple separation of the different motions layers via thresholding. This segmentation-like behaviour, that is desired in many optic flow applications, is a direct consequence of using TV as discontinuity-preserving regulariser. Please note that these flow fields are just one example of the quality of the proposed real-time implementations. Of course, all results presented in our qualitative experiments in the previous chapter can be reproduced with our multigrid methods.

4.5   Summary

In the previous chapter we presented seven prototypes of variational optic flow methods for small displacements and showed that some of them are among the most accurate optic flow methods in the literature. In this chapter we have demonstrated that all these prototypes can be implemented in such a way that they even achieve *real-time performance* for sequences with images of 160×120 pixels. To this end, we discussed different numerical schemes for solving the linear and nonlinear systems that are obtained when minimising the *convex* energy functionals of the different prototypes via their Euler-Lagrange equations.

Thereby we focused in particular on the design of *bidirectional multigrid methods* that are among the fastest numerical schemes for fulfilling this task. In this context we not only derived efficient numerical schemes for all our prototypes, we even succeeded to establish a complete multigrid framework for variational optic flow methods. This framework, that is based on the diffusion and motion tensor notation as well as the discretisation coarse grid approximation approach, allows a systematic construction of efficient hierarchical solvers for all variational methods that are designed in accordance with our theoretical guidelines in Chapter 2.

In our experimental section the advantages of the developed bidirectional multigrid solvers became obvious: Compared to standard numerical methods our implementations achieved speedups of two to four orders of magnitude. Even more efficient solvers such as unidirectional multigrid methods or the successive overrelaxation method were outperformed clearly in most cases. Thereby frame rates of up to 63 dense flow fields per second for sequences of 160×120 allowed for the *first time* in the literature the real-time computation of optic flow fields with variational methods.

Apart from demonstrating the excellent performance of our multigrid implementations we also answered the question if an elliptic or a parabolic strategy is more suitable for minimising the energy functionals of variational methods: We saw that the use of a parabolic strategy does not give any advantages. This strategy required so large time step sizes for an optimal performance that it actually approximated the behaviour of a true *elliptic* strategy.

5   Optic Flow For Large Displacements – Extension

"To err is human, but to really foul things up you need a computer."

- Paul Ehrlich

So far we have restricted ourselves to the design of variational optic flow methods for *small* displacements, i.e. to approaches that use constancy assumptions in their *linearised* form. This restriction allowed us to construct convex energy functionals that offer a *unique* solution and can therefore be minimised by *any* globally convergent algorithm. Moreover, it allowed us to introduce the motion tensor notation that formed the basis of our real-time multigrid framework. However, in particular with respect to real-world sequences where large objects often move faster than a few pixels per frame and thus require the estimation of *large* displacements, such a restriction may not be appropriate. In these cases it may even prevent the approach from estimating the correct displacement field and thus turns from a restriction into a limitation.

In this chapter we *lift this limitation* and *give up the convexity* of the resulting models: We show by the example of an eighth prototype how our previous framework for the design of accurate variational models and the construction of efficient numerical schemes can be extended to the case of *large displacements*. Thereby all steps – the modelling, the minimisation, the discretisation and the numerical scheme – are discussed in detail.

To this end, we start in Section 5.1 with the presentation of our new prototype – an energy functional that is based on constancy assumptions *without* linearisation. Since this functional is no longer convex its minimisation becomes a non-trivial task. This problem is addressed in Section 5.2 where we derive the associated Euler-Lagrange equations and discuss a suitable minimisation strategy. In this context, we also investigate the question of an appropriate extension for our motion tensor notation. After discussing a suitable discretisation in Section 5.3 we continue with a detailed evaluation of the estimation quality of the selected prototype in Section 5.4. In order to allow for a direct comparison to the prototypes for small displacements, the same test scenarios are used as in Chapter 3. This qualitative evaluation is followed by an investigation of efficient numerical schemes in Section 5.5. Thereby, we focus once more on the design of nonlinear bidirectional multigrid methods to solve the obtained nonlinear systems of equations. In this context, we also show how our real-time multigrid framework from Chapter 4 can be extended to variational methods for large displacements. Finally, we conclude this chapter with an evaluation of the proposed numerical scheme. This is done in Section 5.6.

5.1   Prototypes for Large Displacements

As we have already indicated in the introduction to this chapter the design of variational methods for large displacements differs from that of their small-displacement-counterparts:

*In order to handle large displacements one has to refrain from linearising the constancy
assumptions in the model and postpone their linearisation to the minimisation scheme.* As
we will see in the remainder of this chapter, the decision when to linearise the constancy
assumptions – in the model or in the minimisation scheme – has an enormous impact on
both the minimisation strategy and the quality of the resulting flow fields. In order to
demonstrate this, let us start with our first step: the introduction of our eighth prototype.

5.1.1   Prototype H – The Method of Bruhn and Weickert

As our prototype for variational methods for large displacements, we selected the method
of Bruhn and Weickert – a technique that we recently proposed in [BW05]. This method
constitutes an improved variant of the approach of Papenberg *et al.* [PBB$^+$06, BBPW04]
and tackles four important problems at the same time: *varying illumination*, *noise*, *motion
discontinuities* and *large displacements*.

   Such a robustness and flexibility requires the integration of several concepts: Firstly,
our method is based on two different constancy assumptions: the *brightness constancy*
assumption and the *gradient constancy* assumption. Evidently, this allows the handling of
additive illumination changes (if the weight for gradient constancy assumptions is chosen
sufficiently large). Secondly, these assumptions are used in their *original nonlinear form*.
This in turn allows the correct estimation of large displacements: No linearisations are
performed in the model that may spoil the estimation. Thirdly, in order to render our
method more robust with respect to noise and outliers, a *nonquadratic* penaliser function is
applied to *each* of the two constancy assumptions. As we have discussed in Chapter 2 such
a *separate robustification* is more suitable than its joint counterpart if the L_1 norm is used.
Fourthly, our method makes use of an isotropic flow-driven regularisation strategy based
on the total variation (TV). This finally allows the preservation of sharp discontinuities in
the flow field. Combining all these concepts within a single variational framework, the
following energy functional is obtained:

$$E_{\mathrm{BW-2D-LD}}(\mathbf{u}) = \int_{\Omega_2} \left(\widetilde{M}_{14}(D^1 f, D^2 f, \mathbf{u}) + \alpha\, S_4(\nabla f, \nabla \mathbf{u}) \right) d\mathbf{x}$$

$$= \int_{\Omega_2} \Bigg(\underbrace{\Psi_{\mathrm{D}_1}\big(|f(\mathbf{x}+\mathbf{u}) - f(\mathbf{x})|^2\big)}_{\text{robust brightness constancy}} + \gamma\, \underbrace{\Psi_{\mathrm{D}_2}\big(|\nabla f(\mathbf{x}+\mathbf{u}) - \nabla f(\mathbf{x})|^2\big)}_{\text{robust gradient constancy}}$$

$$+ \alpha\, \underbrace{\Psi_{\mathrm{S}}\big(|\nabla_3 u_1|^2 + |\nabla_3 u_2|^2 \big)}_{\text{isotropic flow-driven regulariser}} \Bigg)\, d\mathbf{x}. \tag{5.1}$$

Here, $\widetilde{M}_{13}$ denotes the variant of the data term M_{13} *without* linearisation (brightness and
gradient constancy, separate robustification). Moreover, the three non-quadratic penalisers
Ψ_{D_1}, Ψ_{D_2} and Ψ_{S} are regularised variants of the L_1 norm. The associated small regular-
isation constants are denoted by ϵ_{D_1}, ϵ_{D_2} and ϵ_S. In order to point out the difference to
methods for small displacements we added the abbreviation "LD" which stands for *large
displacements*. Please note that for the sake of simplicity we restrict ourselves to the use
of a spatial regulariser. However, in our comparison to other methods from the literature
in Section 5.6, also results for the corresponding spatiotemporal variant will be presented.

5.2 Minimisation

Following the calculus of variations [Els61] we can minimise the preceding energy functional by solving its Euler–Lagrange equations (*elliptic approach*). As for our prototypes A-G we restrict ourselves only to this strategy and do not consider a minimisation via its steepest descent equations (*parabolic approach*). Once more this decision is motivated by our numerical experiments: They show that also in the case of large displacements, the elliptic approach is more efficient.

5.2.1 The Euler–Lagrange Equations

In the case of small displacements we established a taxonomy of variational optic flow methods based on their Euler–Lagrange equations. We distinguished three cases: *linear* equations, *partly nonlinear* equations and *fully nonlinear* equations. However, since all our models were convex, we implicitly assumed by this classification that equations which belong to one of these three classes have a unique solution. In particular with respect to the fact that energy functionals with constancy assumptions in their original nonlinear form are in general *nonconvex*, this cannot be guaranteed any longer for our prototype H. In order to account for this problem, we have to extend our classification by a fourth class: the class of *fully nonlinear* equations *with multiple solutions*. In the following this class will be referred to as the *nonlinear case III*.

5.2.1.1 The Nonlinear Case III – Prototype H

Let us now derive the Euler–Lagrange equations that are associated with our prototype H. They are given by the following system of coupled partial differential equations

$$
\begin{aligned}
0 = \quad & \Psi'_{\mathrm{D}_1}\big(\,|f(\mathbf{x}+\mathbf{u}) - f(\mathbf{x})|^2\,\big)\,\big(f(\mathbf{x}+\mathbf{u}) - f(\mathbf{x})\big)\,f_{x_1}(\mathbf{x}+\mathbf{u}) \\
& + \gamma\,\Psi'_{\mathrm{D}_2}\big(\,|\nabla f(\mathbf{x}+\mathbf{u}) - \nabla f(\mathbf{x})|^2\,\big)\,\big(f_{x_1}(\mathbf{x}+\mathbf{u}) - f_{x_1}(\mathbf{x})\big)\,f_{x_1 x_1}(\mathbf{x}+\mathbf{u}) \\
& + \gamma\,\Psi'_{\mathrm{D}_2}\big(\,|\nabla f(\mathbf{x}+\mathbf{u}) - \nabla f(\mathbf{x})|^2\,\big)\,\big(f_{x_2}(\mathbf{x}+\mathbf{u}) - f_{x_2}(\mathbf{x})\big)\,f_{x_1 x_2}(\mathbf{x}+\mathbf{u}) \\
& - \alpha\,\mathcal{L}_{\mathrm{NL}}(u_1, u_2),
\end{aligned}
\tag{5.2}
$$

$$
\begin{aligned}
0 = \quad & \Psi'_{\mathrm{D}_1}\big(\,|f(\mathbf{x}+\mathbf{u}) - f(\mathbf{x})|^2\,\big)\,\big(f(\mathbf{x}+\mathbf{u}) - f(\mathbf{x})\big)\,f_{x_2}(\mathbf{x}+\mathbf{u}) \\
& + \gamma\,\Psi'_{\mathrm{D}_2}\big(\,|\nabla f(\mathbf{x}+\mathbf{u}) - \nabla f(\mathbf{x})|^2\,\big)\,\big(f_{x_1}(\mathbf{x}+\mathbf{u}) - f_{x_1}(\mathbf{x})\big)\,f_{x_1 x_2}(\mathbf{x}+\mathbf{u}) \\
& + \gamma\,\Psi'_{\mathrm{D}_2}\big(\,|\nabla f(\mathbf{x}+\mathbf{u}) - \nabla f(\mathbf{x})|^2\,\big)\,\big(f_{x_2}(\mathbf{x}+\mathbf{u}) - f_{x_2}(\mathbf{x})\big)\,f_{x_2 x_2}(\mathbf{x}+\mathbf{u}) \\
& - \alpha\,\mathcal{L}_{\mathrm{NL}}(u_2, u_1)
\end{aligned}
\tag{5.3}
$$

with (reflecting) Neumann boundary conditions. As for the prototypes D and F the *nonlinear* differential operator $\mathcal{L}_{\mathrm{NL}}$ is given by

$$
\mathcal{L}_{\mathrm{NL}}(z(\mathbf{x}), \tilde{z}(\mathbf{x})) = \mathrm{div}\,\big(T(\nabla z(\mathbf{x}), \nabla \tilde{z}(\mathbf{x}))\,\nabla z(\mathbf{x})\big)
\tag{5.4}
$$

with the diffusion tensor

$$\boxed{\text{H - Bruhn/Weickert} \qquad T(\nabla z, \nabla \tilde{z}) = \Psi'_{\text{S}}(|\nabla z|^2 + |\nabla \tilde{z}|^2)\, I}$$

that is evidently nonlinear, since the derivative of the regularised L_1 norm reads

$$\Psi'_{\text{S}}(s^2) = \frac{1}{2\sqrt{s^2 + \epsilon_{\text{S}}^2}} \; . \tag{5.5}$$

Before we can discretise the Euler–Lagrange equations in (5.2)-(5.3), we have to deal with two major problems:

- *Implicit Expressions.* Firstly, due to the use of constancy assumptions without linearisation, terms have appeared in this coupled system of partial differential equations that *implicitly* depend on the desired flow field **u**. Evidently, these terms do not allow a direct discretisation. Therefore we have to find a suitable strategy to resolve them.

- *Local Minima.* Secondly, the underlying energy functional has multiple local minima, since it is *nonconvex*. This in turn means that simple numerical methods such as gradient descent techniques *are no longer globally convergent*: They get trapped in the local minimum that is close to the initialisation. Since all local minima are per definition a correct solution of the Euler–Lagrange equations, we have to develop a minimisation concept that allows to find the solution that corresponds to the global minimum or at least to a reasonable local one.

Apart from solving these two problems it would also be useful to introduce a motion tensor equivalent for variational methods for large displacements. This notation would allow to carry over all numerical strategies that were used before in the context of small displacements. As we will see now, this is strongly connected to the question how the implicit expressions in the Euler–Lagrange equations can be resolved.

5.2.2   Hierarchical Incremental Minimisation

In order to tackle the aforementioned problems, we propose to embed the solution of the Euler–Lagrange equations in an incremental computation based on a coarse-to-fine fixed point iteration. This is done in three steps:

I) *The Fixed Point Iteration.* Firstly, we introduce a fixed point iteration to solve the preceding nonlinear equations. In order to allow for a faster convergence and a better stability than explicit schemes, we use thereby an approach that is semi-implicit in the data and fully implicit in the smoothness term. This yields an iteration instruction that still requires to solve a coupled system of nonlinear equations – however now in *each* iteration step. For the iteration step $k+1$ this system is given by

$$
\begin{aligned}
0 = \quad & \Psi'_{\mathrm{D}_1}\!\big(\ \big|f(\mathbf{x}{+}\mathbf{u}^{k+1}) - f(\mathbf{x})\big|^2\ \big)\,\big(f(\mathbf{x}{+}\mathbf{u}^{k+1}) - f(\mathbf{x})\big)\,f_{x_1}(\mathbf{x}{+}\mathbf{u}^k) \\
+ & \gamma\,\Psi'_{\mathrm{D}_2}\!\big(\ \big|\nabla f(\mathbf{x}{+}\mathbf{u}^{k+1}) - \nabla f(\mathbf{x})\big|^2\ \big)\,\big(f_{x_1}(\mathbf{x}{+}\mathbf{u}^{k+1}) - f_{x_1}(\mathbf{x})\big)\,f_{x_1 x_1}(\mathbf{x}{+}\mathbf{u}^k) \\
+ & \gamma\,\Psi'_{\mathrm{D}_2}\!\big(\ \big|\nabla f(\mathbf{x}{+}\mathbf{u}^{k+1}) - \nabla f(\mathbf{x})\big|^2\ \big)\,\big(f_{x_2}(\mathbf{x}{+}\mathbf{u}^{k+1}) - f_{x_2}(\mathbf{x})\big)\,f_{x_1 x_2}(\mathbf{x}{+}\mathbf{u}^k) \\
- & \alpha\,\mathcal{L}_{\mathrm{NL}}(u_1^{k+1}, u_2^{k+1})
\end{aligned}
\tag{5.6}
$$

$$
\begin{aligned}
0 = \quad & \Psi'_{\mathrm{D}_1}\!\big(\ \big|f(\mathbf{x}{+}\mathbf{u}^{k+1}) - f(\mathbf{x})\big|^2\ \big)\,\big(f(\mathbf{x}{+}\mathbf{u}^{k+1}) - f(\mathbf{x})\big)\,f_{x_2}(\mathbf{x}{+}\ \mathbf{u}^k) \\
+ & \gamma\,\Psi'_{\mathrm{D}_2}\!\big(\ \big|\nabla f(\mathbf{x}{+}\mathbf{u}^{k+1}) - \nabla f(\mathbf{x})\big|^2\ \big)\,\big(f_{x_1}(\mathbf{x}{+}\mathbf{u}^{k+1}) - f_{x_1}(\mathbf{x})\big)\,f_{x_1 x_2}(\mathbf{x}{+}\mathbf{u}^k) \\
+ & \gamma\,\Psi'_{\mathrm{D}_2}\!\big(\ \big|\nabla f(\mathbf{x}{+}\mathbf{u}^{k+1}) - \nabla f(\mathbf{x})\big|^2\ \big)\,\big(f_{x_2}(\mathbf{x}{+}\mathbf{u}^{k+1}) - f_{x_2}(\mathbf{x})\big)\,f_{x_2 x_2}(\mathbf{x}{+}\mathbf{u}^k) \\
- & \alpha\,\mathcal{L}_{\mathrm{NL}}(u_2^{k+1}, u_1^{k+1})\ .
\end{aligned}
\tag{5.7}
$$

II) *The Incremental Computation.* Secondly, we split up the unknown flow field u_1^{k+1} and u_2^{k+1} into the already known part u_1^k and u_2^k and the unknown motion increment du_1^k and du_2^k:

$$
u_1^{k+1} = u_1^k + du_1^k, \qquad u_2^{k+1} = u_2^k + du_2^k.
\tag{5.8}
$$

Since u_1^k and u_2^k are known from the previous step, only the motion increment du_1^k and du_2^k has to be computed in each iteration. This, however, can only be done if all terms are resolved that implicitly depend on this motion increment. To this end, we propose their linearisation via a first order Taylor expansion:

$$
f(\mathbf{x}{+}\mathbf{u}^{k+1}) \approx f(\mathbf{x}{+}\mathbf{u}^k) + f_{x_1}(\mathbf{x}{+}\mathbf{u}^k)\,du_1^k + f_{x_2}(\mathbf{x}{+}\mathbf{u}^k)\,du_2^k
\tag{5.9}
$$

$$
f_{x_1}(\mathbf{x}{+}\mathbf{u}^{k+1}) \approx f_{x_1}(\mathbf{x}{+}\mathbf{u}^k) + f_{x_1 x_1}(\mathbf{x}{+}\mathbf{u}^k)\,du_1^k + f_{x_1 x_2}(\mathbf{x}{+}\mathbf{u}^k)\,du_2^k
\tag{5.10}
$$

$$
f_{x_2}(\mathbf{x}{+}\mathbf{u}^{k+1}) \approx f_{x_2}(\mathbf{x}{+}\mathbf{u}^k) + f_{x_1 x_2}(\mathbf{x}{+}\mathbf{u}^k)\,du_1^k + f_{x_2 x_2}(\mathbf{x}{+}\mathbf{u}^k)\,du_2^k
\tag{5.11}
$$

Please recall that this linearisation has been *intentionally* postponed from the modelling phase in order to allow for a correct handling of large displacements. Apart from resolving all implicit expressions, this linearisation has another advantage: Squared differences of type $|f_*(\mathbf{x} + \mathbf{u}^{k+1}) - f_*(\mathbf{x})|^2$ with f_* being either f, f_{x_1} or f_{x_2} now simplify to the quadratic form

$$
\begin{aligned}
& \big|f_*(\mathbf{x}{+}\mathbf{u}^{k+1}) - f_*(\mathbf{x})\big|^2 \\
\approx\ & \big|f_*(\mathbf{x}{+}\mathbf{u}^k) + f_{*x_1}(\mathbf{x}{+}\mathbf{u}^k)\,du_1^k + f_{*x_2}(\mathbf{x}{+}\mathbf{u}^k)\,du_2^k - f_*(\mathbf{x})\big|^2 \\
=\ & \big|f_{*x_1}(\mathbf{x}{+}\mathbf{u}^k)\,du_1^k + f_{*x_2}(\mathbf{x}{+}\mathbf{u}^k)\,du_2^k + \big(f_*(\mathbf{x}{+}\mathbf{u}^k) - f_*(\mathbf{x})\big)\big|^2 \\
=\ & \big|\big(\mathbf{du}^k\big)^{\!\top}\widetilde{\nabla} f_*(\mathbf{x} + \mathbf{u}^k)\big|^2 \\
=\ & \big(\mathbf{du}^k\big)^{\!\top}\underbrace{\widetilde{\nabla} f_*(\mathbf{x} + \mathbf{u}^k)\widetilde{\nabla} f_*^{\top}(\mathbf{x} + \mathbf{u}^k)}_{J_*(\mathbf{x} + \mathbf{u}^k)}\big(\mathbf{du}^k\big)\ .
\end{aligned}
\tag{5.12}
$$

Here, $\mathbf{du}^k = (du_1^k, du_2^k, 1)^\top$ denotes the spatiotemporal motion increment and $\widetilde{\nabla}$ defines a variant of the ∇ operator where the last component is not a temporal derivative but a temporal difference (which is in fact a simple approximation to the derivative). This shows that even in the case of *large displacements* our systematic notation based on *motion and diffusion tensors* can be used. By defining the new motions tensors for the brightness and the gradient constancy assumption as

$$\widetilde{J_1}^k = \widetilde{\nabla} f(\mathbf{x}+\mathbf{u}^k) \, \widetilde{\nabla} f^\top(\mathbf{x}+\mathbf{u}^k) \tag{5.13}$$

$$\widetilde{J_2}^k = \widetilde{\nabla} f_{x_1}(\mathbf{x}+\mathbf{u}^k) \, \widetilde{\nabla} f_{x_1}^\top(\mathbf{x}+\mathbf{u}^k) + \widetilde{\nabla} f_{x_2}(\mathbf{x}+\mathbf{u}^k) \, \widetilde{\nabla} f_{x_2}^\top(\mathbf{x}+\mathbf{u}^k) \tag{5.14}$$

we can finally write down the partially linearised fixed point iteration step for the motion increment du_1^k and du_2^k:

$$\begin{aligned}
0 = \quad & \Psi'_{\mathrm{D}_1}\big((\mathbf{du}^k)^\top \widetilde{J_1}^k (\mathbf{du}^k)\big)\big(\widetilde{J_1}^k{}_{11} \, du_1^k + \widetilde{J_1}^k{}_{12} \, du_2^k + \widetilde{J_1}^k{}_{13}\big) \\
& + \gamma \, \Psi'_{\mathrm{D}_2}\big((\mathbf{du}^k)^\top \widetilde{J_2}^k (\mathbf{du}^k)\big)\big(\widetilde{J_2}^k{}_{11} \, du_1^k + \widetilde{J_2}^k{}_{12} \, du_2^k + \widetilde{J_2}^k{}_{13}\big) \\
& - \alpha \, \mathcal{L}_{\mathrm{NL}}(u_1^k + du_1^k, u_2^k + du_2^k), \tag{5.15}
\end{aligned}$$

$$\begin{aligned}
0 = \quad & \underbrace{\Psi'_{\mathrm{D}_1}\big((\mathbf{du}^k)^\top \widetilde{J_1}^k (\mathbf{du}^k)\big)\big(\widetilde{J_1}^k{}_{12} \, du_1^k + \widetilde{J_1}^k{}_{22} \, du_2^k + \widetilde{J_1}^k{}_{23}\big)}_{\text{nonlinear}} \\
& + \gamma \, \underbrace{\Psi'_{\mathrm{D}_2}\big((\mathbf{du}^k)^\top \widetilde{J_2}^k (\mathbf{du}^k)\big)\big(\widetilde{J_2}^k{}_{12} \, du_1^k + \widetilde{J_2}^k{}_{22} \, du_2^k + \widetilde{J_2}^k{}_{23}\big)}_{\text{nonlinear}} \\
& - \alpha \, \underbrace{\mathcal{L}_{\mathrm{NL}}(u_2^k + du_2^k, u_1^k + du_1^k)}_{\text{nonlinear}} \ . \tag{5.16}
\end{aligned}$$

One should note that these equations are still nonlinear due to the expressions Ψ'_{D_1}, Ψ'_{D_2} and Ψ'_{S}. However, in contrast to the equations of the original fixed point iteration, they belong to the *nonlinear case II*: They offer a unique solution that can be found by any globally convergent algorithm. This is a direct consequence of the linearisation and the usage of strictly convex penaliser functions (regularised L_1 norm).

After this system has been solved with respect to the motion increments du_1^k and du_2^k, we can update the current overall solution u_1^{k+1} and u_2^{k+1} and therewith the motion tensors $\tilde{J}_1{}^k$ and $\tilde{J}_2{}^k$ for both constancy assumptions. This, however, requires the computation of $f_*(\mathbf{x}+\mathbf{u})$ – a compensation of the image sequence and its derivatives by the already computed motion (current overall solution). What remains to be solved at the next fixed point step is the obtained *difference problem*.

III) *The Coarse-To-Fine Strategy.* Thirdly, in order to tackle the problem of multiple minima, we embed the partly linearised fixed point iteration in (5.15)–(5.16) into a coarse-to-fine framework. During this procedure we use representations of the image sequence f smoothed at different scales to update the motion increment du_1^k and du_2^k. Starting the iteration with a very smooth version of f that we define as f^1, we employ successively less smoothed representations of f in subsequent iteration

steps. This can for instance be achieved by convolving f with Gaussians of decreasing standard deviation. If we connect the variable k not only to the iteration index but also to the current scale of the image sequence f, only the computation of the motion tensors changes with respect to the non-hierarchical fixed point equations in (5.15)-(5.16). They are now given by

$$\widetilde{J}_1^{\,k} \;=\; \widetilde{\nabla} f^k(\mathbf{x}+\mathbf{u}^k)\,\widetilde{\nabla} f^{k,\top}(\mathbf{x}+\mathbf{u}^k), \tag{5.17}$$

$$\widetilde{J}_2^{\,k} \;=\; \widetilde{\nabla} f_{x_1}^k(\mathbf{x}+\mathbf{u}^k)\,\widetilde{\nabla} f_{x_1}^{k,\top}(\mathbf{x}+\mathbf{u}^k) + \widetilde{\nabla} f_{x_2}^k(\mathbf{x}+\mathbf{u}^k)\,\widetilde{\nabla} f_{x_2}^{k,\top}(\mathbf{x}+\mathbf{u}^k) \tag{5.18}$$

where f^k denotes the smoothed variant of the image sequence that is used at the current iteration k.

In contrast to the unidirectional multigrid methods that we have used for accelerating the solution of linear and nonlinear systems of equations in Chapter 4, this coarse-to-fine strategy has also a second, even more important purpose: *the avoidance of local minima*. Since the underlying energy functional is nonconvex, the initialisation decides to which (local) minimum the fixed point iteration converges. A hierarchical proceeding offers hereby the advantage, that local minima with sufficiently small spatial extent vanish at coarser scales and can thus be avoided. Two examples for such a coarse-to-fine minimisation strategy is shown in Figure 5.1. As one can see, the strategy may successfully find the global minimum (a) or at least avoids sufficiently small local minima (b).
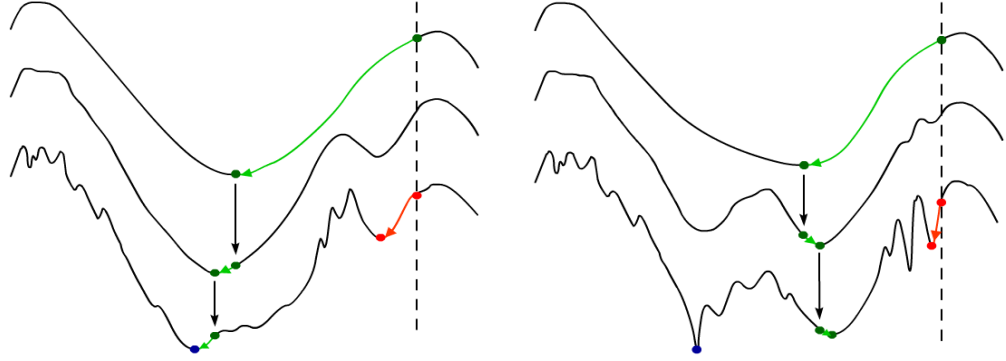
Fig. 5.1: **Red:** Non-hierarchical minimisation. **Green:** Coarse-to-fine minimisation. **Blue:** Global minimum. **(a) Left:** Global optimum found. **(b) Right:** Useful local minimum found.

One should note that in the case of our incremental fixed point iteration only the scale has to be refined between the different fixed point steps. The initialisation, however, is automatically accomplished by modifying the problem (cf. step II).

Let us once again point out the goals of the three previous steps: While the coarse-to-fine strategy can easily be identified with a unidirectional multigrid scheme that helps to *avoid local minima* during the optimisation, the incremental computation via a fixed point iteration is nothing else than an approximation of the original nonlinear constancy assumptions in the data term by means of a *series of linearised ones*. Although the coarse-to-fine technique works well in most of the cases, one should note that there exists no convergence proof for it. However, there are already convergence results for a simplified optic problem using such coarse-to-fine strategies [LC01]

5.3 Discretisation

In the previous section, we have proposed to minimise the nonconvex energy functional of our prototype H by means of an incremental coarse-to-fine fixed point iteration. Thereby at *each iteration step* a coupled system of nonlinear partial differential equations has to be solved in order to determine the current motion increment. Let us now discuss how this system can be discretised and how the presented coarse-to-fine framework can be implemented. To this end, we make use of the discretisation scheme in Section 3.3.1 and introduce additionally the following abbreviations:

$$\mathbf{du}_l^{k,\mathbf{h}} \ := \ \left(\left[du_l^{k,\mathbf{h}} \right]_{1,1}, \ldots, \left[du_l^{k,\mathbf{h}} \right]_{N_{x_1}^{\mathbf{h}}, N_{x_2}^{\mathbf{h}}} \right)^\top \quad (l = 1,2), \tag{5.19}$$

$$\widetilde{\mathbf{j}}\mathbf{l}_{mn}^{k,\mathbf{h}} \ := \ \left(\left[\widetilde{J}_l{}^{k,\mathbf{h}}_{mn} \right]_{1,1}, \ldots, \left[\widetilde{J}_l{}^{k,\mathbf{h}}_{mn} \right]_{N_{x_1}^{\mathbf{h}}, N_{x_2}^{\mathbf{h}}} \right)^\top \quad (l = 1,2 \ ; \ m,n = 1,2,3), \tag{5.20}$$

$$\widetilde{J}_l{}^{k,\mathbf{h}}_{mn} \ := \ \mathrm{diag}\left(\left(\widetilde{\mathbf{j}}\mathbf{l}_{mn}^{k,\mathbf{h}} \right)^\top \right) \quad (l = 1,2 \ ; \ m,n = 1,2,3). \tag{5.21}$$

5.3.1 The Discrete Fixed Point Step

Structure. We are now in the position to discretise the partially linearised fixed point iteration step in (5.15)–(5.16). The obtained *nonlinear* system of equations that has to be solved at each fixed point step with respect to the $2N^{\mathbf{h}}$ unknowns $\mathbf{du}_1^{k,\mathbf{h}}$ and $\mathbf{du}_2^{k,\mathbf{h}}$ is then given by

$$
\left(
\underbrace{
\begin{pmatrix}
F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}})\, \widetilde{J}_1{}^{k,\mathbf{h}}_{11} & F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}})\, \widetilde{J}_1{}^{k,\mathbf{h}}_{12} \\
F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}})\, \widetilde{J}_1{}^{k,\mathbf{h}}_{12} & F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}})\, \widetilde{J}_1{}^{k,\mathbf{h}}_{22}
\end{pmatrix}
}_{\text{nonlinear operator}}
\right.
$$

$$
+\gamma
\underbrace{
\begin{pmatrix}
F_{2,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}})\, \widetilde{J}_2{}^{k,\mathbf{h}}_{11} & F_{2,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}})\, \widetilde{J}_2{}^{k,\mathbf{h}}_{12} \\
F_{2,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}})\, \widetilde{J}_2{}^{k,\mathbf{h}}_{12} & F_{2,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}})\, \widetilde{J}_2{}^{k,\mathbf{h}}_{22}
\end{pmatrix}
}_{\text{nonlinear operator}}
$$

$$
-\alpha
\underbrace{
\begin{pmatrix}
L_{\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{u}_1^{k,\mathbf{h}}+\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{u}_2^{k,\mathbf{h}}+\mathbf{du}_2^{k,\mathbf{h}}) & 0 \\
0 & L_{\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{u}_1^{k,\mathbf{h}}+\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{u}_2^{k,\mathbf{h}}+\mathbf{du}_2^{k,\mathbf{h}})
\end{pmatrix}
}_{\text{nonlinear operator}}
\left.
\vphantom{\begin{pmatrix}1\\1\end{pmatrix}}
\right)
\begin{pmatrix}
\mathbf{du}_1^{k,\mathbf{h}} \\
\mathbf{du}_2^{k,\mathbf{h}}
\end{pmatrix}
+
$$

$$
\left(
\underbrace{
\begin{pmatrix}
F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}})\, \widetilde{\mathbf{j}}\mathbf{1}_{13}^{k,\mathbf{h}} \\
F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}})\, \widetilde{\mathbf{j}}\mathbf{1}_{23}^{k,\mathbf{h}}
\end{pmatrix}
}_{\text{nonlinear operator}}
+\gamma
\underbrace{
\begin{pmatrix}
F_{2,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}})\, \widetilde{\mathbf{j}}\mathbf{2}_{13}^{k,\mathbf{h}} \\
F_{2,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}})\, \widetilde{\mathbf{j}}\mathbf{2}_{23}^{k,\mathbf{h}}
\end{pmatrix}
}_{\text{nonlinear operator}}
\right.
$$

$$
-\alpha
\underbrace{
\begin{pmatrix}
L_{\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{u}_1^{k,\mathbf{h}}+\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{u}_2^{k,\mathbf{h}}+\mathbf{du}_2^{k,\mathbf{h}}) & 0 \\
0 & L_{\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{u}_1^{k,\mathbf{h}}+\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{u}_2^{k,\mathbf{h}}+\mathbf{du}_2^{k,\mathbf{h}})
\end{pmatrix}
}_{\text{nonlinear operator}}
\begin{pmatrix}
\mathbf{u}_1^{k,\mathbf{h}} \\
\mathbf{u}_2^{k,\mathbf{h}}
\end{pmatrix}
\left.
\vphantom{\begin{pmatrix}1\\1\end{pmatrix}}
\right)
=
\underbrace{
\begin{pmatrix}
\mathbf{0} \\
\mathbf{0}
\end{pmatrix}
}_{\text{rhs}}.
\tag{5.22}
$$

Here, the nonlinear factors $\Psi'_{D_1}\big((\mathbf{du}^k)^\top \widetilde{J}_1^{\ k}\,(\mathbf{du}^k)\big)$ and $\Psi'_{D_2}\big((\mathbf{du}^k)^\top \widetilde{J}_2^{\ k}\,(\mathbf{du}^k)\big)$ result in the nonlinear operators

$$F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}}) := \mathrm{diag}\Big(\Psi'_{D_1}\big(\big[\mathbf{du}^{k,\mathbf{h}}\big]_{1,1}^\top \big[\tilde{J}_1^{k,\mathbf{h}}\big]_{1,1}\big[\mathbf{du}^{k,\mathbf{h}}\big]_{1,1}\big), \ldots, \tag{5.23}$$

$$\ldots, \Psi'_{D_1}\big(\big[\mathbf{du}^{k,\mathbf{h}}\big]_{N_{x_1}^{\mathbf{h}}, N_{x_2}^{\mathbf{h}}}^\top \big[\widetilde{J}_1^{\ k,\mathbf{h}}\big]_{N_{x_1}^{\mathbf{h}}, N_{x_2}^{\mathbf{h}}}\big[\mathbf{du}^{k,\mathbf{h}}\big]_{N_{x_1}^{\mathbf{h}}, N_{x_2}^{\mathbf{h}}}\big)\Big),$$

$$F_{2,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}}) := \mathrm{diag}\Big(\Psi'_{D_2}\big(\big[\mathbf{du}^{k,\mathbf{h}}\big]_{1,1}^\top \big[\tilde{J}_2^{k,\mathbf{h}}\big]_{1,1}\big[\mathbf{du}^{k,\mathbf{h}}\big]_{1,1}\big), \ldots, \tag{5.24}$$

$$\ldots, \Psi'_{D_2}\big(\big[\mathbf{du}^{k,\mathbf{h}}\big]_{N_{x_1}^{\mathbf{h}}, N_{x_2}^{\mathbf{h}}}^\top \big[\widetilde{J}_2^{\ k,\mathbf{h}}\big]_{N_{x_1}^{\mathbf{h}}, N_{x_2}^{\mathbf{h}}}\big[\mathbf{du}^{k,\mathbf{h}}\big]_{N_{x_1}^{\mathbf{h}}, N_{x_2}^{\mathbf{h}}}\big)\Big).$$

As we have indicated before, this system of equations has the same structure as the discrete Euler–Lagrange equations of our prototypes F and G (*nonlinear case II*): A unique solution exists, all terms form a single nonlinear operator and the right hand side is homogeneous. However, in the case of our prototype H this system of equations only describes a small part of a complex hierarchical minimisation strategy to solve the associated Euler–Lagrange equations. It constitutes only one out of many fixed point steps that have to be performed within the complete incremental coarse-to-fine framework.

Coupling. In order to allow for the selection of a suitable numerical scheme in Section 5.5, let us briefly discuss the different types of couplings that occur in the equation system of each fixed point step. As in the nonlinear case II, four different types of couplings are present:

- The *explicit point coupling* between $\mathbf{du}_1^{\mathbf{h}}$ and $\mathbf{du}_2^{\mathbf{h}}$
  $\big(\,$via $F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}})\,\widetilde{J}_{1\,12}^{\ k,\mathbf{h}}$ and $F_{2,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}})\,\widetilde{J}_{2\,12}^{\ k,\mathbf{h}}\,\big)$.

- The *implicit point coupling* between $\mathbf{du}_1^{\mathbf{h}}$ and $\mathbf{du}_2^{\mathbf{h}}$
  $\big(\,$via the joint argument of $F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}})$ and $F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{du}_2^{k,\mathbf{h}})\,\big)$.

- The *explicit isotropic neighbourhood coupling* within $\mathbf{du}_1^{k,\mathbf{h}}$ and $\mathbf{du}_2^{k,\mathbf{h}}$
  $\big(\,$via $L_{\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{u}_1^{k,\mathbf{h}}+\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{u}_2^{k,\mathbf{h}}+\mathbf{du}_2^{k,\mathbf{h}})\,\big)$.

- The *implicit point coupling* between $\mathbf{du}_1^{\mathbf{h}}$ and $\mathbf{du}_2^{\mathbf{h}}$
  $\big(\,$via the joint argument of $L_{\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{u}_1^{k,\mathbf{h}}+\mathbf{du}_1^{k,\mathbf{h}}, \mathbf{u}_2^{k,\mathbf{h}}+\mathbf{du}_2^{k,\mathbf{h}})\,\big)$.

Therefore, also in the case of our prototype H it may be useful to consider block relaxation techniques for solving the different equation systems (coupled point relaxation, alternating line relaxation). Which solvers are most suitable and how they can be implemented efficiently will be discussed in Section 5.5.

5.3.2 The Discrete Coarse-To-Fine Fixed Point Iteration

After we have discussed the discretisation of the fixed point step, let us now turn to the question how the coarse-to-fine framework can be implemented. For this purpose, the literature offers us two different strategies: *multiresolution techniques* [BA96, MP98b]

that downsample all relevant data and thus make use of different resolution levels and *scale-space focusing methods* [AWS99] that keep the resolution unchanged but consider the problem at different smoothness scales (similar to the original continuous idea). While the first approach is much more efficient from a computational point of view, the second one is easier to implement: Since the resolution is kept fixed, one does not have to care about the aliasing problem.

Concrete Example. Evidently, we are interested in extending our real-time framework from the previous chapter to variational methods for large displacements. Therefore, we have to realise the coarse-to-fine strategy with the more efficient *multiresolution technique*. In this context, three decisions have to be made:

I) *Motion Compensation Strategy.* Firstly, at each resolution level, the image sequence has to be compensated by the already computed flow field $\mathbf{u}_1^{k,\mathbf{h}}$ and $\mathbf{u}_2^{k,\mathbf{h}}$ in order to compute the new entries for $\tilde{J}_1^{k,\mathbf{h}}$ and $\tilde{J}_2^{k,\mathbf{h}}$. This requires a *motion compensation strategy* – an algorithm that distorts an image in accordance with a given vector field. As proposed in [MP98a, BBPW04] this is realised by a *backward registration* approach based on *bilinear interpolation*.

II) *Intergrid Transfer Operators.* Secondly, both a *restriction* and a *prolongation* operator has to be defined. While the restriction operator is required to transfer the image sequence f to coarser resolution levels, the prolongation operator is needed for interpolating the current overall solution $\mathbf{u}_1^{k,\mathbf{h}}$ and $\mathbf{u}_2^{k,\mathbf{h}}$ back to the next finer grid. As in the case of uni- and bidirectional multigrid methods this is accomplished by means of *area-based averaging* and *area-based interpolation* [BWKS05].

III) *Downsampling Factor.* Thirdly, a downsampling factor $\eta \in (0,1)$ has to be chosen that states by which factor *each dimension* is reduced from one resolution level to the next (the actual resolution ratio between two levels is then given by η^2). In our case we selected η from the interval [0.5,0.7]. Values from this interval are in general a good compromise between computational efficiency (the smaller the value for η the faster the computation) and the quality of the minimisation (the larger the value for η the better the results).

Relation to Warping Methods. One should note that the incorporation of the partly linearised fixed point iteration into a multiresolution strategy yields the well-known *warping technique* [BAHH92, BA96, MP98a]. Warping denotes the distortion of the image sequence which is required for the compensation for the already computed motion. So far this technique has only been justified on an *algorithmic basis*: In general, it was argued that it makes sense to embed optic flow approaches for small displacements into a coarse-to-fine framework, since large displacements become smaller at coarser levels and thus allow for an accurate estimation with linearised model assumptions. This, of course, is true. However, as we have seen, this warping strategy can also be derived as a hierarchical fixed point iteration for minimising the energy functional of a variational approach for large displacements, i.e. for an energy functional based on constancy assumptions without linearisation. This in turn, provides a *theoretical justification* of the warping technique.

5.4 Qualitative Experiments

In the previous section we have shown by the example of our prototype H how energy functionals for large displacements can be minimised. Let us now make use of this knowledge and evaluate the estimation quality of such methods with respect to different motion scenarios. In order to allow for a direct comparison of the achieved results to those obtained by our prototypes for small displacements, the same experiments with synthetic and real-world sequences are performed as in Chapter 3. Thereby, we restrict ourselves to comparisons with the advanced prototypes F and G – these approaches yielded the qualitatively best results in our first evaluation.

5.4.1 Synthetic Sequences

Let us begin our evaluation with experiments on synthetic image sequences. To this end, we make use of the same structure as in Chapter 3: Starting from relatively simple global motion patterns such as divergence and translations we increase the complexity of the scenes step by step.

5.4.1.1 Experiment I: Globally Divergent Motion

In our first experiment we study the performance of the prototype H with respect to globally divergent motion. This is done by the example of the *Yosemite* sequence *without* clouds. As one can see from the obtained results in Table 5.1, it is evidently very useful to postpone the linearisation of the constancy assumptions to the minimisation scheme: Although our prototype H relies only on spatial regularisation and thus cannot benefit from the temporally smooth motion pattern, it still shows an estimation quality that is comparable to the one of our spatiotemporal prototype G. This similar performance of both prototypes is also confirmed by the corresponding flow fields in Figure 5.2: While the prototype H allows a slightly better estimation of the large mountain in the foreground – this area requires the computation of relatively large displacements (up to 5.5 pixels) – the prototype G gives better results in the remaining regions. At the end of this section, we will show that an extension of the prototype H to the spatiotemporal setting combines the strength of both approaches: Thus, even average angular errors below one degree become possible.

Tab. 5.1: **Experiment I**: Performance of the most accurate prototypes for synthetic sequences with globally divergent motion. We optimised the parameters σ_{spat}, σ_{temp}, α, ρ and γ with respect to the average angular error (AAE).

Yosemite Sequence without Clouds

prototype	method	σ_{spat}	α	other parameters	AAE
F	Bruhn *et al.* 2-D, SD	1.95	17	ρ=1.3	2.27°
G	Papenberg *et al.* 3-D, SD	1.95	7	γ=1.0, σ_{temp}=0.80	1.45°
H	Bruhn/Weickert 2-D, LD	0.90	130	γ=7.4	1.58°
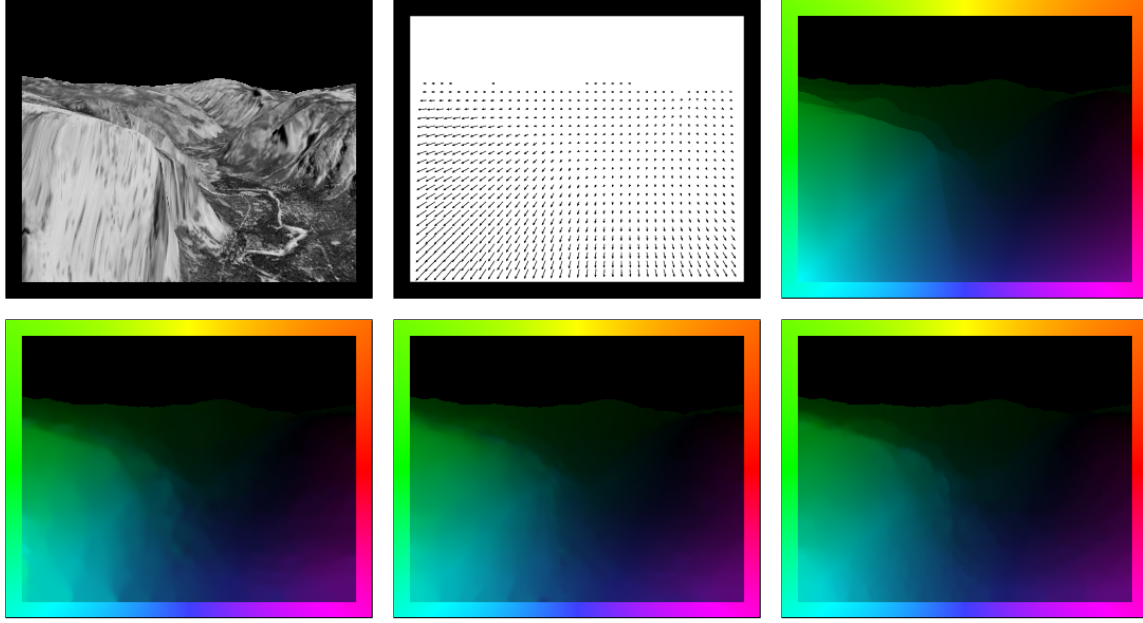
Fig. 5.2: **Experiment I**: Performance of the most accurate prototypes for synthetic sequences with globally divergent motion. **(a) Top Left:** Frame 8 of the *Yosemite* sequence *without* clouds of size 316×256 (grey-scale). **(b) Top Centre:** Ground truth (vector plot) **(c) Top Right:** Ground truth (colour plot). **(d) Bottom Left:** Prototype F (Bruhn *et al.* 2-D, SD). **(e) Bottom Centre:** Prototype G (Papenberg *et al.* 3-D, SD). **(f) Bottom Right:** Prototype H (Bruhn/Weickert 2-D, LD).

Fig. 5.3: **Experiment II**: Performance of the most accurate prototypes for synthetic sequences with discontinuities, outliers and globally divergent motion. **(a) Top Left:** Frame 10 of the *Office* sequence of size 200×200 (grey-scale variant). **(b) Top Centre:** Ground truth (vector plot) **(c) Top Right:** Ground truth (colour plot). **(d) Bottom Left:** Prototype F (Bruhn *et al.* 2-D, SD). **(e) Bottom Centre:** Prototype G (Papenberg *et al.* 3-D, SD). **(f) Bottom Right:** Prototype H (Bruhn/Weickert 2-D, LD).

Fig. 5.4: **Experiment III**: Performance of the most accurate prototypes for synthetic sequences with discontinuities, outliers and translational motion with static background. **(a) Top Left:** Frame 150 of the *New Marble* sequence of size 512×384 (grey-scale variant). **(b) Top Centre:** Ground truth (vector plot) **(c) Top Right:** Ground truth (colour plot). **(d) Bottom Left:** Prototype F (Bruhn *et al.* 2-D, SD). **(e) Bottom Centre:** Prototype G (Papenberg *et al.* 3-D, SD). **(f) Bottom Right:** Prototype H (Bruhn/Weickert 2-D, LD).

Fig. 5.5: **Experiment IV**: Performance of the most accurate prototypes for synthetic sequences with discontinuities, outliers and translational motion with dynamic background (camera motion). **(a) Top Left:** Frame 10 of the *Street* sequence of size 200×200 (grey-scale variant). **(b) Top Centre:** Ground truth (vector plot) **(c) Top Right:** Ground truth (colour plot). **(d) Bottom Left:** Prototype F (Bruhn *et al.* 2-D, SD). **(e) Bottom Centre:** Prototype G (Papenberg *et al.* 3-D, SD). **(f) Bottom Right:** Prototype H (Bruhn/Weickert 2-D, LD).

5.4.1.2   Experiment II: Discontinuities, Outliers, Globally Divergent Motion

In our second experiment we use the *Office* sequence to investigate the performance of the prototype H with respect to global divergent motion, outliers and discontinuities. Since for this sequence the maximum displacements are in the order of 1.5 pixels, it is not surprising that the obtained results shown in Table 5.2 are not in favour of our prototype H. Moreover, as we have already seen in the context of our qualitative benchmark for small displacements, spatial regularisers at not very effective in suppressing strong temporal aliasing artifacts and preserving discontinuities at the same time. This is also reflected in the corresponding flow fields depicted in Figure 5.3. While our spatial prototypes either suppress artifacts (prototype F) or preserve discontinuities (prototype H), the spatiotemporal prototype G allows to find a suitable compromise.

Tab. 5.2: **Experiment II**: Performance of the most accurate prototypes for synthetic sequences with discontinuities, outliers and globally divergent motion. We optimised the parameters σ_{spat}, σ_{temp}, α, ρ and γ with respect to the average angular error (AAE).

Office Sequence

prototype	method	σ_{spat}	α	other parameters	AAE
F	Bruhn *et al.* 2-D, SD	0.80	17	ρ=2.5	4.31°
G	Papenberg *et al.* 3-D, SD	1.30	7	γ=2.4, σ_{temp}=0.10	3.26°
H	Bruhn/Weickert 2-D, LD	0.80	47	γ=1.9	4.30°

5.4.1.3   Experiment III: Discontinuities, Translational Motion with Static Background

Let us now investigate the performance of our prototype H with respect to translational motion with a *static background*. This is done in our third experiment by the example of the *New Marble* sequence. As one can see from the obtained results in Table 5.3, our approach for large displacements (prototype H) yields the lowest angular error of all three prototypes. Evidently, it is useful for this sequence to rely on constancy assumptions without linearisation. This observation is confirmed by the computed flow fields shown in Figure 5.3: In the case of the prototype H the displacements in the upper right corner of both marble blocks are estimated more precisely compared to the other two prototypes. This has a simple explanation: Since these areas are the parts of the blocks that are closest to the camera, their motion causes the largest displacements (2.5 pixels).
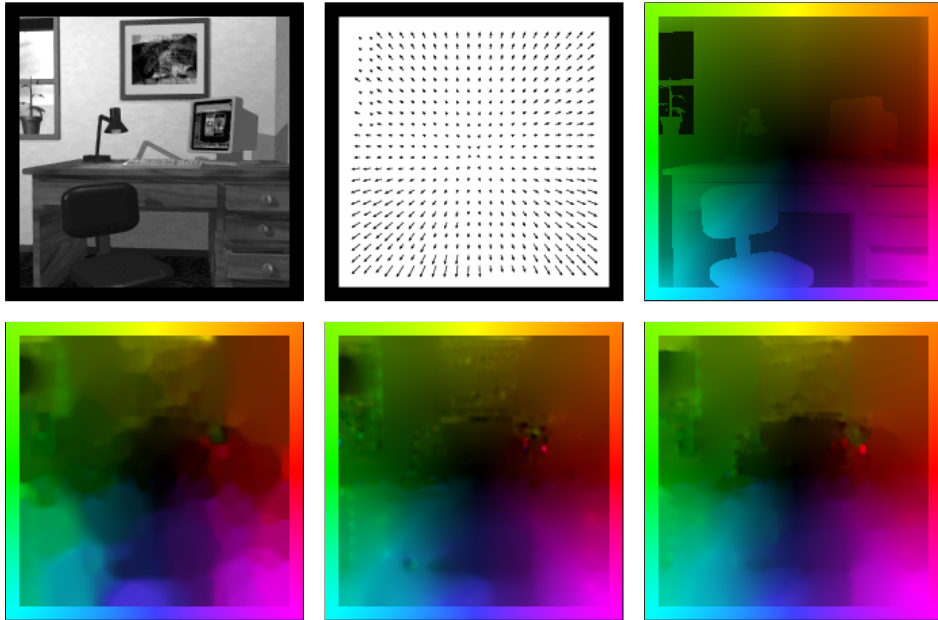
Tab. 5.3: **Experiment III**: Performance of the most accurate prototypes for synthetic sequences with discontinuities and translational motion with static background. We optimised the parameters σ_{spat}, σ_{temp}, α, ρ and γ with respect to the average angular error (AAE).

New Marble Sequence

prototype	method	σ_{spat}	α	other parameters	AAE
F	Bruhn *et al.* 2-D, SD	0.50	27	ρ=0.1	0.93°
G	Papenberg *et al.* 3-D, SD	0.50	19	γ=1.0, σ_{temp}=0.10	0.66°
H	Bruhn/Weickert 2-D, LD	0.00	44	γ=0.7	0.60°

5.4.1.4 Experiment IV: Discontinuities, Outliers, Translational Motion with Moving Background

In our fourth experiment we study the estimation quality of the prototype H with respect to translational motion in front of a *moving background*. This is done in Table 5.4, where the computed average angular errors for the *Street* sequence are listed. Once more, our prototype for large displacements (prototype H) gives the best result. The reason for this can be seen from the corresponding flow fields in Figure 5.4. Although the prototype H cannot completely suppress the outliers – this requires once more a spatiotemporal regulariser – it yields by far the most accurate estimation of the car. This in turn is not surprising, since as in the previous experiment this motion is related to the largest displacements (4.5 pixels).
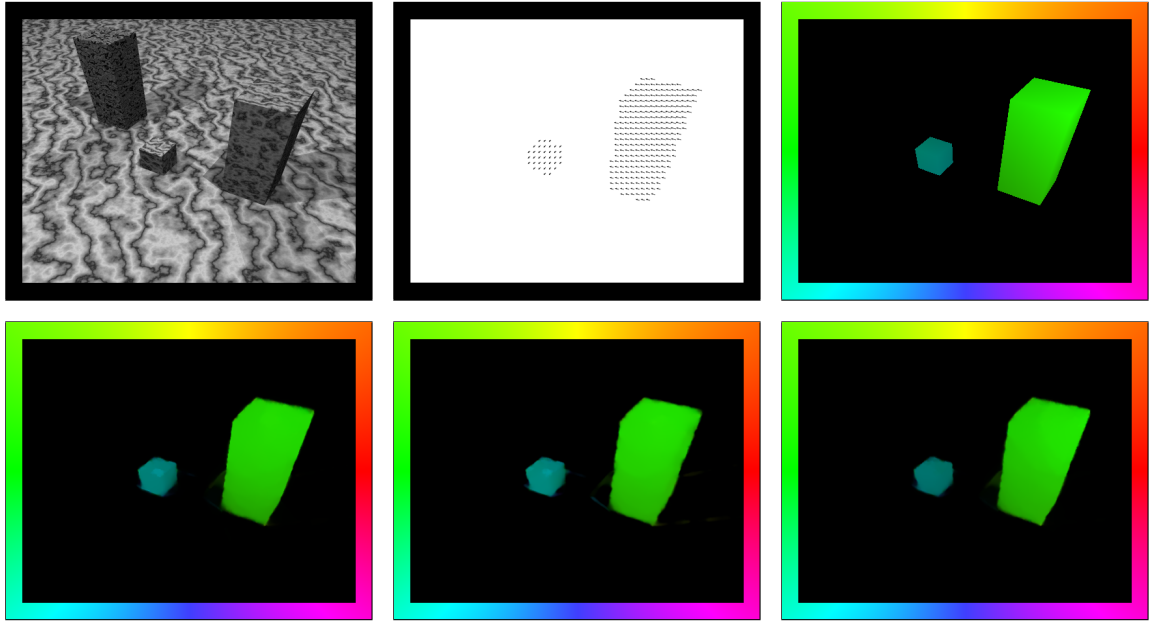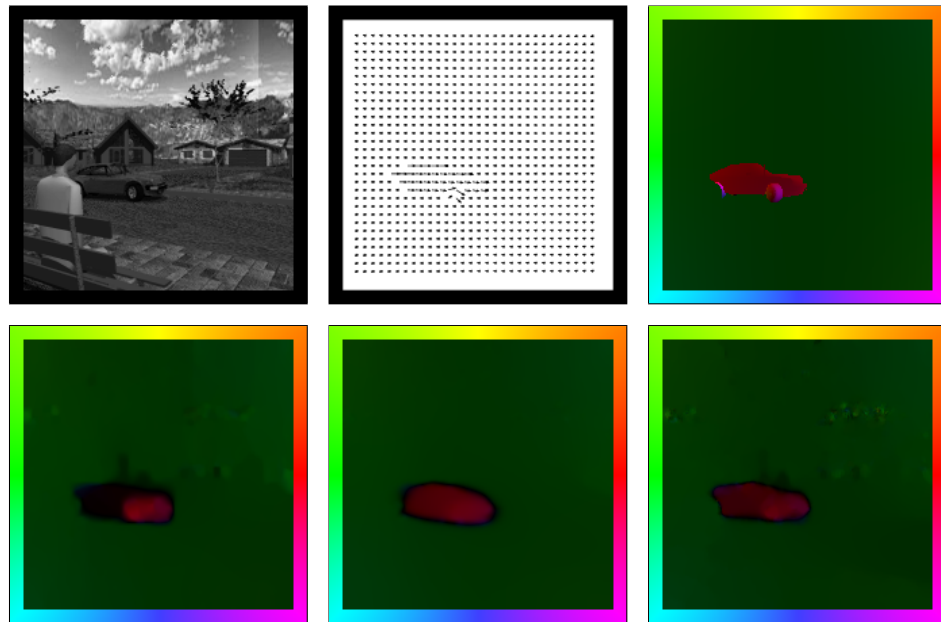
Tab. 5.4: **Experiment IV**: Performance of the most accurate prototypes for synthetic sequences with discontinuities and translational motion with dynamic background (camera motion). We optimised the parameters σ_{spat}, σ_{temp}, α, ρ and γ with respect to the average angular error (AAE).

Street Sequence

prototype	method	σ_{spat}	α	other parameters	AAE
F	Bruhn *et al.* 2-D, SD	0.90	10	ρ=0.9	4.30°
G	Papenberg *et al.* 3-D, SD	1.60	11	γ=2.1, σ_{temp}=0.10	3.21°
H	Bruhn/Weickert 2-D, LD	0.80	187	γ=16.0	3.16°

5.4.1.5 Experiment V: Discontinuities, Translational and Divergent Motion, Varying Illumination

Our fifth experiment is dedicated to the estimation of translational and divergent motion under varying illumination. To this end, we used the *Yosemite* sequence *with* clouds. As in the case of its variant without cloudy sky, this sequence requires the estimation of displacements in the order of up to 5.5 pixels. Expectedly, the achieved results in Table 5.6 show the usefulness of postponing the linearisation of the constancy assumptions to the minimiation scheme. This is also confirmed by the corresponding flow fields in Figure 5.6. In contrast to the prototype G that already linearises the constancy assumptions in the model, the large mountain in the foreground is estimated more precisely. This observation is in accordance with our first experiment where the same sequence without cloudy sky was used.

Tab. 5.5: **Experiment V**: Performance of the most accurate prototypes for synthetic sequences with discontinuities, translational and divergent motion as well as varying illumination. We optimised the parameters σ_{spat}, σ_{temp}, α, ρ and γ with respect to the average angular error (AAE).

Yosemite Sequence with Clouds

prototype	method	σ_{spat}	α	other parameters	AAE
F	Bruhn *et al.* 2-D, SD	1.60	24	ρ=1.4	5.74°
G	Papenberg *et al.* 3-D, SD	2.10	93	γ=1050, σ_{temp}=0.50	2.78°
H	Bruhn/Weickert 2-D, LD	0.90	160	γ=16.5	2.42°

Fig. 5.6: **Experiment V**: Performance of the most accurate prototypes for synthetic sequences with discontinuities, translational and divergent motion as well as varying illumination. **(a) Top Left:** Frame 8 of the *Yosemite* sequence *with* clouds of size 316×256 (grey-scale). **(b) Top Centre:** Ground truth (vector plot) **(c) Top Right:** Ground truth (colour plot). **(d) Bottom Left:** Prototype F (Bruhn *et al.* 2-D, SD). **(e) Bottom Centre:** Prototype G (Papenberg *et al.* 3-D, SD). **(f) Bottom Right:** Prototype H (Bruhn/Weickert 2-D, LD).
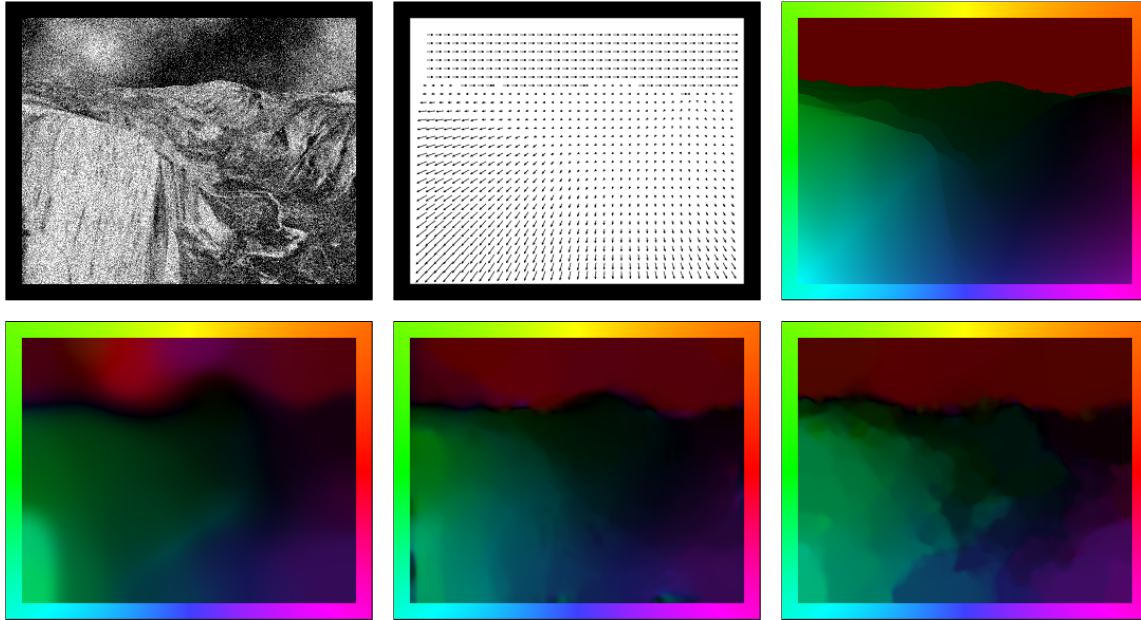
Fig. 5.7: **Experiment VI:** Performance of the most accurate prototypes for synthetic sequences with Gaussian noise. **(a) Top Left:** Frame 8 of the *Yosemite* sequence *with* clouds of size 316×256 (grey-scale). Gaussian noise of $\sigma_n = 40$ was added. **(b) Top Centre:** Ground truth (vector plot) **(c) Top Right:** Ground truth (colour plot). **(d) Bottom Left:** Prototype F (Bruhn *et al.* 2-D, SD). **(e) Bottom Centre:** Prototype G (Papenberg *et al.* 3-D, SD). **(f) Bottom Right:** Prototype H (Bruhn/Weickert 2-D, LD).

5.4.1.6 Experiment VI: Discontinuities, Translational and Divergent Motion, Varying Illumination, Gaussian Noise

In the sixth and last experiment with synthetic image sequences we evaluate the performance of our prototype H with respect to noise. To this end, we considered the same noisy variants of the *Yosemite* sequence *with* clouds as for the corresponding experiment for small displacements (Gaussian noise of standard deviation $\sigma_n = 10, 20$ and 40). The obtained results for all noise levels are presented in Table 5.6. They show the high robustness of our prototype H with respect to noise. However, they also show that a spatiotemporal regulariser (prototype G) can even be slightly more effective in handling the corrupted data. This is also reflected in the computed flow fields for $\sigma_n = 40$ (Figure 5.7). In contrast to the spatiotemporal prototype G that tends to oversmooth the flow field, the prototype H preserves the discontinuities in a better way. However, since it also adapts to noise it yields an undesired segmentation-like result. Evidently, for optimal performance both concepts – constancy assumptions without linearisation and spatiotemporal regularisation – have to be combined. As we have demonstrated in [PBB$^+$06] such a combination even allows to obtain average angular errors of below five degrees for a noise with standard deviation of $\sigma_n = 40$.

Tab. 5.6: **Experiment VI:** Performance of the most accurate prototypes for synthetic sequences with Gaussian noise. We optimised the parameters σ_{spat}, σ_{temp}, α, ρ and γ with respect to the average angular error (AAE).

Yosemite Sequence with Clouds - Noise Level $\sigma_n = 0$

prototype	method	σ_{spat}	α	other parameters	AAE
F	Bruhn *et al.* 2-D, SD	1.60	24	ρ=1.4	5.74°
G	Papenberg *et al.* 3-D, SD	2.10	60	γ=1050, σ_{temp}=0.50	2.78°
H	Bruhn/Weickert 2-D, LD	0.90	160	γ=16.5	2.42°

Yosemite Sequence with Clouds - Noise Level $\sigma_n = 10$

prototype	method	σ_{spat}	α	other parameters	AAE
F	Bruhn *et al.* 2-D, SD	1.80	20	ρ=2.3	7.96°
G	Papenberg *et al.* 3-D, SD	3.10	39	γ=1050, σ_{temp}=0.40	3.95°
H	Bruhn/Weickert 2-D, LD	1.45	71	γ=9.0	4.13°

Yosemite Sequence with Clouds - Noise Level $\sigma_n = 20$

prototype	method	σ_{spat}	α	other parameters	AAE
F	Bruhn *et al.* 2-D, SD	2.10	21	ρ=3.0	10.71°
G	Papenberg *et al.* 3-D, SD	3.80	36	γ=1050, σ_{temp}=0.65	5.58°
H	Bruhn/Weickert 2-D, LD	1.45	61	γ=6.0	5.97°

Yosemite Sequence with Clouds - Noise Level $\sigma_n = 40$

prototype	method	σ_{spat}	α	other parameters	AAE
F	Bruhn *et al.* 2-D, SD	2.40	12	ρ=15.4	15.31°
G	Papenberg *et al.* 3-D, SD	5.00	25	γ=1050, σ_{temp}=0.70	8.29°
H	Bruhn/Weickert 2-D, LD	2.15	61	γ=6.0	9.63°

5.4.2 Real-World Sequences

After we have investigated the performance of our prototype for large displacements with respect to synthetic image sequences, let us now turn to the evaluation of its estimation quality for real-world image data.

5.4.2.1 Experiment I: Discontinuities, Rotational Motion

In our first experiment on real-world data we address the problem of rotational motion. This is done by the example of the *Rubik* sequence. As one can see from the computed flow fields in Figure 5.8, the estimation quality of our prototype H is between the one of the prototypes F and G: Discontinuities at the cube and at the top of the disc are well-preserved (even better than in the case of our prototype F), while discontinuities between the disc and the static floor are not respected (not as good as in the case of our prototype G). This however, is not surprising. Since the maximum displacements for this sequence are in the order of 1.5 pixels, our prototype H can hardly benefit from the fact that it is based on constancy assumptions in their original nonlinear form.
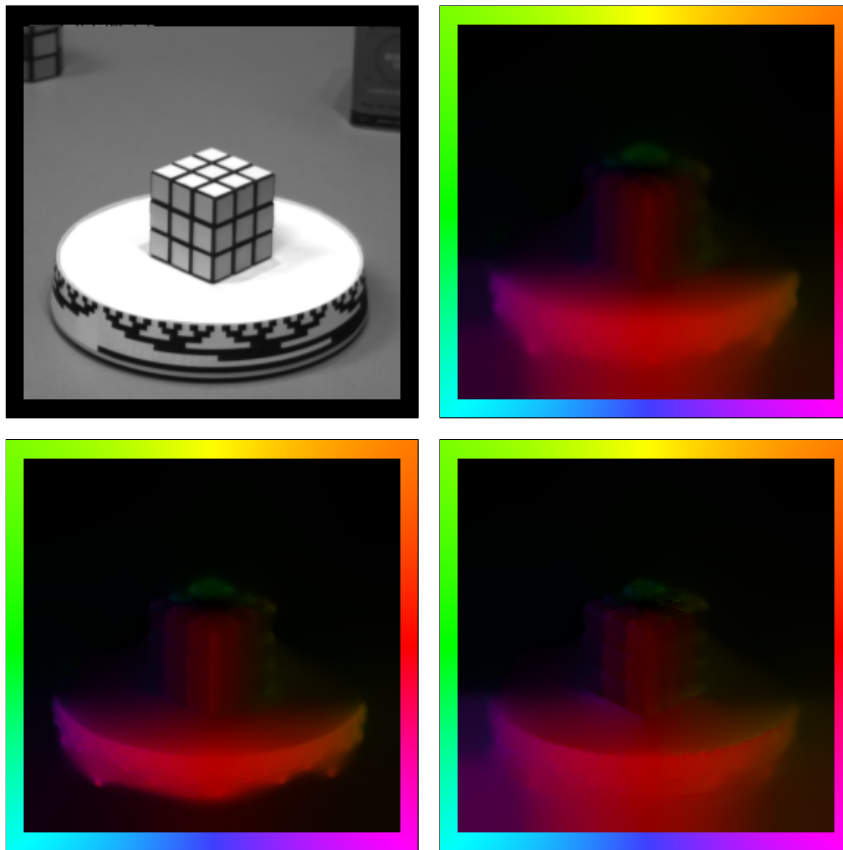
Fig. 5.8: **Experiment I**: Performance of the most accurate prototypes for real-world sequences with discontinuities and rotational motion. **(a) Top Left:** Frame 10 of the *Rubic* sequence of size 256×240 (grey-scale). **(b) Top Right:** Prototype F (Bruhn *et al.* 2-D, SD). **(c) Bottom Left:** Prototype G (Papenberg *et al.* 3-D, SD). **(d) Bottom Right:** Prototype H (Bruhn/Weickert 2-D, LD).

5.4.2.2 Experiment II: Discontinuities, Multiple Translational Motions, Interlacing

In our second experiment on real-world sequences we evaluate the performance of our prototype H with respect to a classical traffic scene: the *Rheinhafen* sequence. Apart from multiple motions and interlacing this sequence requires the estimation of the largest displacements of all test scenes (up to 9 pixels per frame). Expectedly, the computed flow fields depicted in Figure 5.10 make the advantages of constancy assumptions without linearisation explicit: While the prototypes F and G underestimate the motion of the large white van due to the use of linearised constancy assumptions, our prototype H allows for a correct estimation. Even the motion of the remaining cars is now estimated more accurately and thus better distinguishable from the static background (they also move faster than computed by the prototypes F and G). One should note that in contrast to the same experiment for small displacements in Chapter 3, the colour representation of the flow fields for F and G was adapted such that they became comparable to the one of our prototype H. The full saturation is now achieved for displacements of 9 pixels (instead of for displacements of 2 pixels).
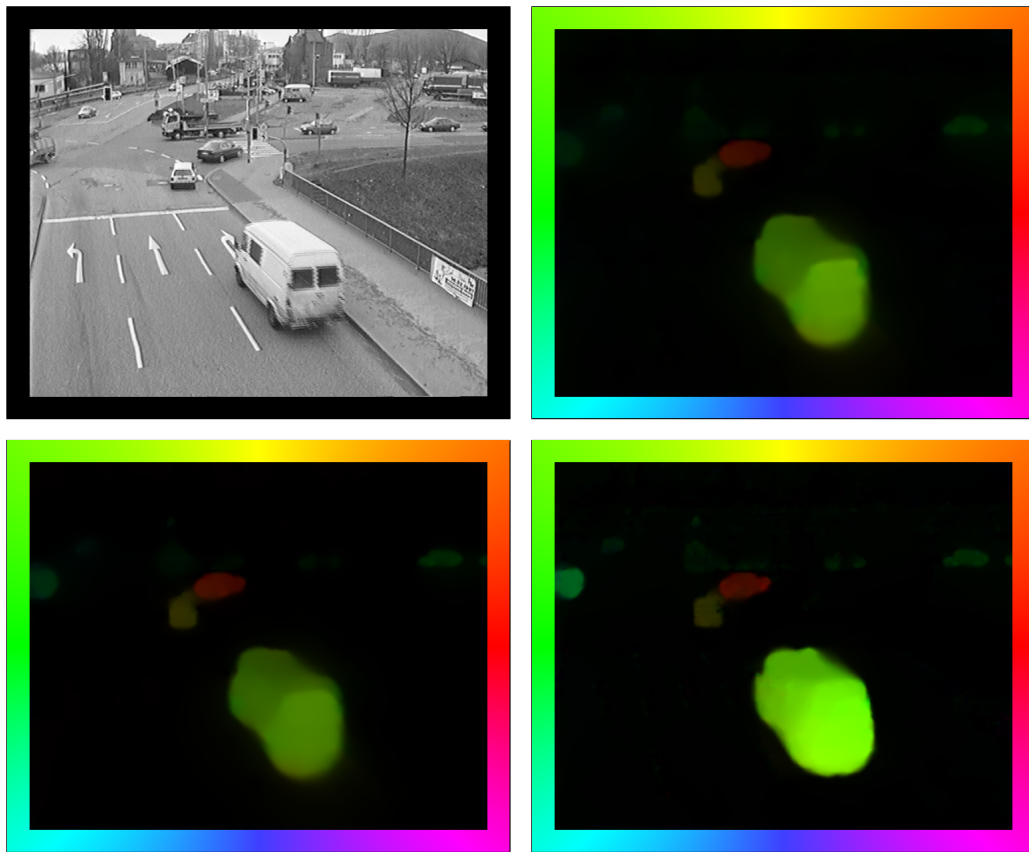
Fig. 5.9: **Experiment II**: Performance of the most accurate prototypes for real-world sequences with discontinuities, multiple translational motions and interlacing. **(a) Top Left:** Frame 1130 of the interlaced *Rheinhafen* sequence of size 688×565 (grey-scale). **(b) Top Right:** Prototype F (Bruhn *et al.* 2-D, SD). **(c) Bottom Left:** Prototype G (Papenberg *et al.* 3-D, SD). **(d) Bottom Right:** Prototype H (Bruhn/Weickert 2-D, LD).

5.4.2.3   Experiment III: Discontinuities, Multiple Translational Motions, Occlusions, Interlacing

In our third and last experiment we evaluate the performance of our prototype H with respect to occlusions. To this end, we consider another popular traffic scene: the *Ettlinger Tor* sequence. As in the previous experiment, the computed flow fields show clearly the advantages of constancy assumptions without linearisation (Figure 5.10). In particular, the fastest moving objects – the bus in the foreground and the car that overtakes this bus – are estimated more precisely by our prototype H. For our prototypes F and G the corresponding displacements of up to 6 pixels per frame are already too large to allow for an accurate estimation: Parts of the bus that are relatively close to the occlusion created by the lamp post suffer from a consequent underestimation. This tendency becomes even more explicit in the fact that the overtaking car is almost ignored completely.
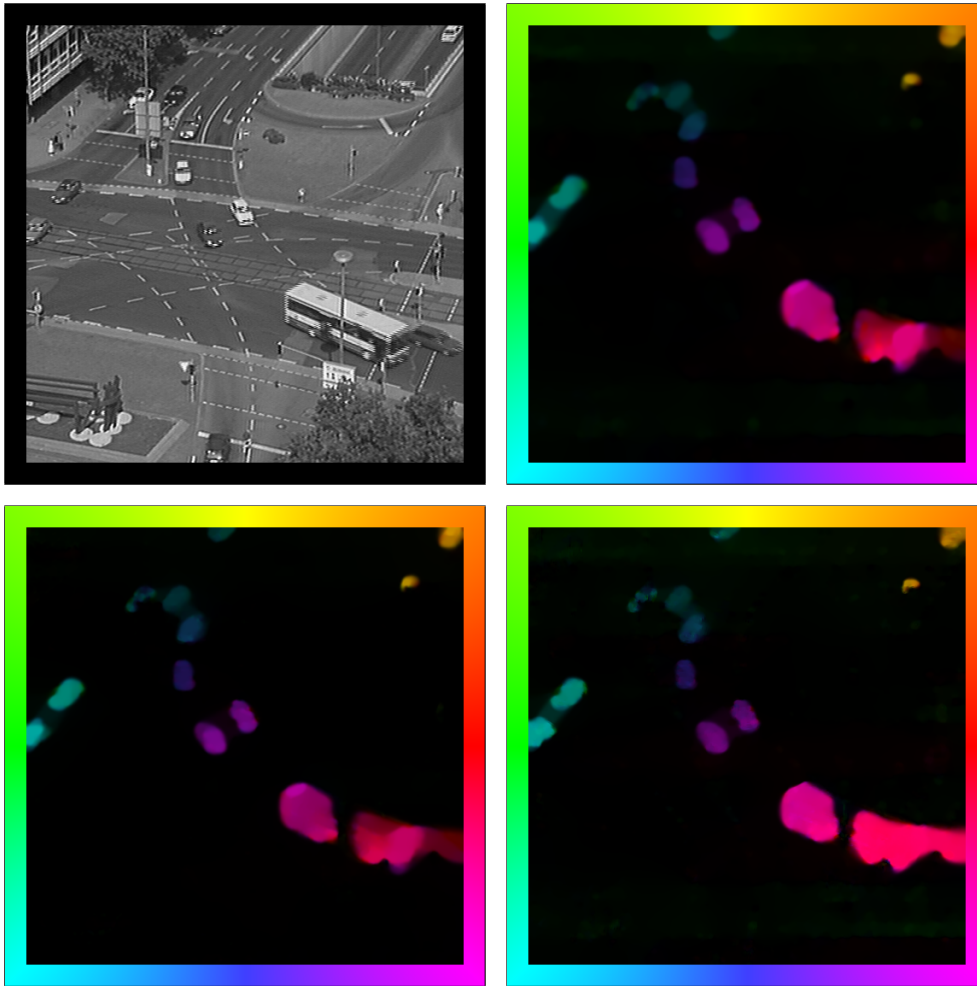
Fig. 5.10: **Experiment III**: Performance of the most accurate prototypes for real-world sequences with discontinuities, multiple translational motions, occlusions, and interlacing. **(a) Top Left:** Frame 24 of the *Ettlinger Tor* sequence of size 512×512 (grey-scale). **(b) Top Right:** Prototype F (Bruhn *et al.* 2-D, SD). **(c) Bottom Left:** Prototype G (Papenberg *et al.* 3-D, SD). **(d) Bottom Right:** Prototype H (Bruhn/Weickert 2-D, LD).

Tab. 5.7: **Comparison To The Literature I:** Comparison between results from the literature with 100 % density and our prototypes. All data refer to the *Yosemite* sequence with cloudy sky. SD = small displacements. LD = large displacements. HC = constancy assumptions on higher derivatives. NQ-D = non-quadratic (robust) data term. NQ-S = non-quadratic smoothness term or similar strategy (preserves discontinuities). M = multiscale strategy, warping (can handle large displacements). S = segmentation strategy. AAE = average angular error.

Yosemite Sequence with Clouds

technique		HC	NQ-D	NQ-S	3-D	M	S	AAE
Horn/Schunck, orig.	[BFB94]	-	-	-	-	-	-	31.69°
Singh, step 1	[BFB94]	-	-	-	-	-	-	15.28°
Anandan	[BFB94]	-	-	-	-	-	-	13.36°
Singh, step 2	[BFB94]	-	-	-	-	-	-	10.44°
Nagel	[BFB94]	-	-	✓	-	-	-	10.22°
Horn/Schunck, mod.	[BFB94]	-	-	-	-	-	-	9.78°
Uras *et al.*	[BFB94]	-	-	-	-	-	-	8.94°
Prototype A		-	-	✓	-	-	-	7.17°
Liu *et al.*	[LCR98]	-	-	✓	✓	-	-	6.85 °
Prototype B		-	-	✓	-	-	-	6.44°
Prototype E		-	-	✓	-	-	-	6.42°
Prototype D		-	-	✓	-	-	-	6.32°
Prototype C		-	-	-	-	-	-	6.28°
Prototype F (2-D, SD)		-	✓	✓	-	-	-	5.74°
Alvarez *et al.*	[AWS00]	-	-	✓	-	✓	-	5.53°
Mémin/Pérez	[MP98a]	-	✓	✓	-	✓	-	5.38°
Prototype F (3-D, SD)		-	✓	✓	✓	-	-	5.18°
Farnebäck	[Far01]	✓	-	-	✓	-	✓	4.84°
Mémin/Pérez	[MP98b]	-	✓	✓	-	✓	-	4.69°
Prototype F (3-D, LD)	[BWS05]	-	✓	✓	✓	✓	-	**4.17°**
Wu *et al.*	[WaJCL98]	-	-	-	✓	-	-	3.54°
Prototype G (2-D, SD)		✓	✓	✓	✓	-	-	3.50°
Prototype G (3-D, SD)		✓	✓	✓	✓	-	-	2.78°
Teng *et al.*	[TLCH05]	-	✓	✓	✓	✓	-	2.70°
Prototype H (2-D, LD)	[BW05]	✓	✓	✓	-	✓	-	**2.42°**
Amiaz/Kiryati	[AK05a]	✓	✓	✓	-	✓	✓	2.04°
Prototype G (3-D, LD)	[PBB$^+$06]	✓	✓	✓	✓	✓	-	**1.78°**
Amiaz/Kiryati	[AK05b]	✓	✓	✓	-	✓	✓	1.73°
Prototype H (3-D, LD)	[BW05]	✓	✓	✓	✓	✓	-	**1.72°**

5.4.3   Comparison To the Literature

Let us finally compare our results to the most accurate results from the literature. To this end, we updated the corresponding tables for the *Yosemite* sequence *with* and *without* clouds from Chapter 3 and added entries for the average angular errors obtained by the spatial and spatiotemporal variant of our prototype H. Moreover, in order to allow for a better comparison between methods for large and small displacements, we also included results of variants of our prototypes F and G for large displacements (marked with "LD").

Tab. 5.8: **Comparison To The Literature II:** Comparison between results from the literature with 100 % density and our prototypes. All data refer to the *Yosemite* sequence without cloudy sky. SD = small displacements. LD = large displacements. HC = constancy assumptions on higher derivatives. NQ-D = non-quadratic (robust) data term. NQ-S = non-quadratic smoothness term (preserves discontinuities). M = multiscale strategy, warping (can handle large displacements). S = segmentation strategy. AAE = average angular error.

Yosemite Sequence without Clouds

technique		HC	NQ-D	NQ-S	3-D	M	S	AAE
Black/Anandan	[BA96]	-	✓	✓	-	✓	-	4.56°
Black	[Bla94]	-	✓	✓	✓	✓	-	3.52°
Szeliski/Coughlan	[SC94]	-	-	-	✓	✓	-	2.45°
Prototype E		-	-	✓	-	-	-	2.65°
Prototype A		-	-	-	-	-	-	2.64°
Prototype B		-	-	✓	-	-	-	2.64°
Prototype D		-	-	✓	-	-	-	2.58°
Prototype C		-	-	✓	-	-	-	2.57°
Prototype G (2-D, SD)		✓	✓	✓	-	-	-	2.30°
Black/Jepson	[BJ96]	-	✓	✓	-	✓	-	2.29°
Prototype F (2-D, SD)		-	✓	✓	-	-	-	2.27°
Middendorf	[Mid03]	-	-	✓	✓	-	-	2.27°
Ju *et al.*	[JBJ96]	-	✓	✓	-	✓	✓	2.16°
Bab Hadiashar/Suter	[BHS98]	-	-	-	-	-	-	2.05°
Lai/Vemuri	[LV98]	-	-	-	-	-	-	1.99°
Mémin/Pérez	[MP02]	-	✓	✓	-	✓	✓	1.58°
Prototype H (2-D, LD)	[BW05]	✓	✓	✓	✓	-	-	**1.58°**
Teng *et al.*	[TLCH05]	-	✓	✓	✓	✓	-	1.52°
Roth/Black	[RB05]	-	✓	✓	✓	✓	-	1.47°
Prototype F (3-D, SD)		-	✓	✓	✓	-	-	1.46°
Prototype G (3-D, SD)		✓	✓	✓	✓	-	-	1.45°
Farnebäck	[Far00]	✓	-	-	✓	-	✓	1.40°
Liu *et al.*	[LCR98]	-	-	✓	✓	-	-	1.39 °
Farnebäck	[Far01]	✓	-	-	✓	-	✓	1.14°
Prototype F (3-D, LD)	[BWS05]	-	✓	✓	✓	✓	-	**1.02°**
Prototype G (3-D, LD)	[PBB^{+}06]	✓	✓	✓	✓	✓	-	**0.98°**
Prototype H (3-D, LD)	[BW05]	✓	✓	✓	✓	✓	-	**0.98°**

5.4.3.1   Comparison I: Yosemite with Clouds

Table 5.7 compares results obtained for the *Yosemite* sequence *with* clouds. It is evident that our prototypes perform favourably. In particular, the spatiotemporal variant of our prototype H and the "long displacement variant" of our prototype G yield excellent results. Thereby, with an average angular error of 1.72°, the former one achieves the *currently most accurate result* in the literature. In this context one should note that even the good results of the approaches of Amiaz and Kiryati are related to the high accuracy of our prototypes: These methods are based on the embedding of the spatial variant of prototype G into a level set formulation [AK05a, AK05b].

5.4.3.2 Comparison II: Yosemite without Clouds

The comparison in Table 5.8 for the *Yosemite* sequence without clouds confirms our previous findings. With average angular errors below and around $1°$, variants of all three prototypes yield *more accurate results* than any other approach proposed in the literature so far. This underlines once more that a systematic design of variational methods allows to construct models that are transparent and highly accurate *at the same time*.

5.5 Solvers

Having designed some of the best performing variational techniques in the literature, let us now discuss how these methods can be accelerated in such a way that they can be used for time-critical applications. Evidently, this requires to combine the multiresolution technique from Section 5.3 with an efficient numerical solver for the resulting nonlinear systems of equations. In the following we show that also in this case our real-time multigrid framework from Chapter 4 can be applied. To this end, we derive a suitable bidirectional nonlinear multigrid method for our prototype H – a full approximation scheme [Bra77].

5.5.1 The Nonlinear Case III

General Problem. Let us start by giving a general definition of our problem. In the following we are interested in solving a *series* of n nonlinear equation systems given by

$$
\begin{aligned}
A^{1,\mathbf{h}}(\mathbf{x}^{1,\mathbf{h}}) &= \mathbf{b}^{1,\mathbf{h}} \\
\vdots \quad \vdots \quad &\quad \vdots \\
A^{n,\mathbf{h}}(\mathbf{x}^{n,\mathbf{h}}) &= \mathbf{b}^{n,\mathbf{h}}.
\end{aligned}
\tag{5.25}
$$

where each of the equation systems $k = 1, .., n$ consists of a nonlinear operator $A^{k,\mathbf{h}}(\mathbf{x}^{k,\mathbf{h}})$ and a right hand side $b^{k,\mathbf{h}}$. Thereby, each operator $A^{k,\mathbf{h}}(\mathbf{x}^{\mathbf{h}})$ can be decomposed into

$$
A^{k,\mathbf{h}}(\mathbf{x}^{k,\mathbf{h}}) = B^{k,\mathbf{h}}(\mathbf{x}^{k,\mathbf{h}})\,\mathbf{x}^{k,\mathbf{h}} + \mathbf{c}^{k,\mathbf{h}}(\mathbf{x}^{k,\mathbf{h}})
\tag{5.26}
$$

where $B^{\mathbf{h}}(\mathbf{x}^{k,\mathbf{h}})$ and $\mathbf{c}^{\mathbf{h}}(\mathbf{x}^{k,\mathbf{h}})$ are nonlinear operators such that for any fixed $\mathbf{x}^{k,\mathbf{h}}$ the evaluated expression $B^{k,\mathbf{h}}(\mathbf{x}^{k,\mathbf{h}})$ is a *symmetric and positive definite* $2N^{k,\mathbf{h}} \times 2N^{k,\mathbf{h}}$ matrix and $\mathbf{c}^{k,\mathbf{h}}(\mathbf{x}^{k,\mathbf{h}})$ is a vector of size $2N^{k,\mathbf{h}}$. Here, the number of unknowns for a certain equation system k is given by $N^{k,\mathbf{h}} = \lceil \eta^{2(n-k)} N^{n,\mathbf{h}} \rceil$, where η is a downsampling factor in the interval $(0, 1)$. Furthermore, one should note that *each* of the nonlinear equation systems in (5.25) has a *unique* solution (cf. Subsection 5.2.2 step II).

Relation to the Nonlinear Cases I and II. Evidently, the nonlinear case III is closely related to nonlinear cases I and II. In fact, each nonlinear equation system of the series of equation systems that we consider in (5.25) is a specific instance of of our general problem for the nonlinear cases I and II (cf. (4.56) and (4.57)). This in turn makes the development of efficient numerical solvers straightforward: *Any numerical scheme that we have developed in the context of the nonlinear cases I and II can also be used in the context of our nonlinear case III – even our bidirectional multigrid method*. Let us demonstrate this useful relation by the example of our prototype H.

Concrete Example. In our concrete example we are interested in solving the nonlinear equation systems that are associated to an incremental coarse-to-fine fixed point iteration applied to the discrete Euler–Lagrange equations of our prototype H. To this end, let us first identify the different parts of the corresponding nonlinear equation systems in (5.22) with those of our general problem in (5.25)–(5.26). This can be done as follows:

$$
\begin{aligned}
&\Bigg(\underbrace{ \begin{pmatrix} F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}},\mathbf{du}_2^{k,\mathbf{h}})\,\widetilde{J}_{1\,11}^{\;k,\mathbf{h}} & F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}},\mathbf{du}_2^{k,\mathbf{h}})\,\widetilde{J}_{1\,12}^{\;k,\mathbf{h}} \\ F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}},\mathbf{du}_2^{k,\mathbf{h}})\,\widetilde{J}_{1\,12}^{\;k,\mathbf{h}} & F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}},\mathbf{du}_2^{k,\mathbf{h}})\,\widetilde{J}_{1\,22}^{\;k,\mathbf{h}} \end{pmatrix} \\
&+\gamma \begin{pmatrix} F_{2,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}},\mathbf{du}_2^{k,\mathbf{h}})\,\widetilde{J}_{2\,11}^{\;k,\mathbf{h}} & F_{2,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}},\mathbf{du}_2^{k,\mathbf{h}})\,\widetilde{J}_{2\,12}^{\;k,\mathbf{h}} \\ F_{2,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}},\mathbf{du}_2^{k,\mathbf{h}})\,\widetilde{J}_{2\,12}^{\;k,\mathbf{h}} & F_{2,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}},\mathbf{du}_2^{k,\mathbf{h}})\,\widetilde{J}_{2\,22}^{\;k,\mathbf{h}} \end{pmatrix} \\
&-\alpha \begin{pmatrix} L_{\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{u}_1^{k,\mathbf{h}}+\mathbf{du}_1^{k,\mathbf{h}},\mathbf{u}_2^{k,\mathbf{h}}+\mathbf{du}_2^{k,\mathbf{h}}) & 0 \\ 0 & L_{\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{u}_1^{k,\mathbf{h}}+\mathbf{du}_1^{k,\mathbf{h}},\mathbf{u}_2^{k,\mathbf{h}}+\mathbf{du}_2^{k,\mathbf{h}}) \end{pmatrix} }_{B^{k,\mathbf{h}}(\mathbf{x}^{k,\mathbf{h}})} \Bigg) \underbrace{\begin{pmatrix} \mathbf{du}_1^{k,\mathbf{h}} \\ \mathbf{du}_2^{k,\mathbf{h}} \end{pmatrix}}_{\mathbf{x}^{k,\mathbf{h}}} +
\end{aligned}
$$

$$
\begin{aligned}
&\Bigg(\underbrace{ \begin{pmatrix} F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}},\mathbf{du}_2^{k,\mathbf{h}})\,\widetilde{\mathbf{j}}_{\mathbf{1}\,13}^{\;k,\mathbf{h}} \\ F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}},\mathbf{du}_2^{k,\mathbf{h}})\,\widetilde{\mathbf{j}}_{\mathbf{1}\,23}^{\;k,\mathbf{h}} \end{pmatrix} +\gamma \begin{pmatrix} F_{2,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}},\mathbf{du}_2^{k,\mathbf{h}})\,\widetilde{\mathbf{j}}_{\mathbf{2}\,13}^{\;k,\mathbf{h}} \\ F_{2,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}},\mathbf{du}_2^{k,\mathbf{h}})\,\widetilde{\mathbf{j}}_{\mathbf{2}\,23}^{\;k,\mathbf{h}} \end{pmatrix} \\
&-\alpha \begin{pmatrix} L_{\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{u}_1^{k,\mathbf{h}}+\mathbf{du}_1^{k,\mathbf{h}},\mathbf{u}_2^{k,\mathbf{h}}+\mathbf{du}_2^{k,\mathbf{h}}) & 0 \\ 0 & L_{\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{u}_1^{k,\mathbf{h}}+\mathbf{du}_1^{k,\mathbf{h}},\mathbf{u}_2^{k,\mathbf{h}}+\mathbf{du}_2^{k,\mathbf{h}}) \end{pmatrix} \begin{pmatrix} \mathbf{u}_1^{k,\mathbf{h}} \\ \mathbf{u}_2^{k,\mathbf{h}} \end{pmatrix} }_{\mathbf{c}^{k,\mathbf{h}}(\mathbf{x}^{k,\mathbf{h}})} \Bigg) = \underbrace{\begin{pmatrix} 0 \\ 0 \end{pmatrix}}_{\mathbf{b}^{k,\mathbf{h}}}.
\end{aligned}
$$
(5.27)

As one can easily verify each of these systems belongs in fact to the nonlinear case II: all terms are nonlinear and the right hand side is zero. However, before we can make use of our numerical schemes for this case, we have to reformulate these equation systems in point-based notation. This allows us to make the effect of the nonlinear operators $F_{1,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}},\mathbf{du}_2^{k,\mathbf{h}})$ and $F_{2,\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{du}_1^{k,\mathbf{h}},\mathbf{du}_2^{k,\mathbf{h}})$ and the nonlinear differential operator $L_{\mathrm{NL}}^{k,\mathbf{h}}(\mathbf{u}_1^{k,\mathbf{h}}+\mathbf{du}_1^{k,\mathbf{h}},\mathbf{u}_2^{k,\mathbf{h}}+\mathbf{du}_2^{k,\mathbf{h}})$ on $\mathbf{du}_1^{k,\mathbf{h}}$, $\mathbf{du}_2^{k,\mathbf{h}}$, $\mathbf{u}_1^{k,\mathbf{h}}$ and $\mathbf{u}_2^{k,\mathbf{h}}$ explicit and thus to write down the iteration instructions for our different numerical schemes. In point-based notation these nonlinear equation systems are given by

$$
\begin{aligned}
&\left(\left[\Psi_{\mathrm{D}_1}'^{k,\mathbf{h}} \right]_{i,j} \left[\widetilde{J}_{1,11}^{k,\mathbf{h}} \right]_{i,j} + \gamma \left[\Psi_{\mathrm{D}_2}'^{k,\mathbf{h}} \right]_{i,j} \left[\widetilde{J}_{2,11}^{k,\mathbf{h}} \right]_{i,j} \right) \left[du_1^{k,\mathbf{h}} \right]_{i,j} \\
&+ \left(\left[\Psi_{\mathrm{D}_1}'^{k,\mathbf{h}} \right]_{i,j} \left[\widetilde{J}_{1,12}^{k,\mathbf{h}} \right]_{i,j} + \gamma \left[\Psi_{\mathrm{D}_2}'^{k,\mathbf{h}} \right]_{i,j} \left[\widetilde{J}_{2,12}^{k,\mathbf{h}} \right]_{i,j} \right) \left[du_2^{k,\mathbf{h}} \right]_{i,j} \\
&- \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)} \frac{\left(\left[\Psi_{\mathrm{S}}'^{k,\mathbf{h}} \right]_{\tilde{i},\tilde{j}} + \left[\Psi_{\mathrm{S}}'^{k,\mathbf{h}} \right]_{i,j} \right)}{2} \frac{\left(\left[du_1^{k,\mathbf{h}} \right]_{\tilde{i},\tilde{j}} - \left[du_1^{k,\mathbf{h}} \right]_{i,j} \right)}{h_l^2} \\
&+ \left(\left[\Psi_{\mathrm{D}_1}'^{k,\mathbf{h}} \right]_{i,j} \left[\widetilde{J}_{1,13}^{k,\mathbf{h}} \right]_{i,j} + \gamma \left[\Psi_{\mathrm{D}_2}'^{k,\mathbf{h}} \right]_{i,j} \left[\widetilde{J}_{2,13}^{k,\mathbf{h}} \right]_{i,j} \right) \\
&- \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)} \frac{\left(\left[\Psi_{\mathrm{S}}'^{k,\mathbf{h}} \right]_{\tilde{i},\tilde{j}} + \left[\Psi_{\mathrm{S}}'^{k,\mathbf{h}} \right]_{i,j} \right)}{2} \frac{\left(\left[u_1^{k,\mathbf{h}} \right]_{\tilde{i},\tilde{j}} - \left[u_1^{k,\mathbf{h}} \right]_{i,j} \right)}{h_l^2} = 0
\end{aligned}
$$
(5.28)

$$
\left(\left[\Psi_{\mathrm{D}_1}^{\prime\,k,\mathbf{h}} \right]_{i,j} \left[\widetilde{J}_{1,12}^{k,\mathbf{h}} \right]_{i,j} + \gamma \left[\Psi_{\mathrm{D}_2}^{\prime\,k,\mathbf{h}} \right]_{i,j} \left[\widetilde{J}_{2,12}^{k,\mathbf{h}} \right]_{i,j} \right) \left[du_1^{k,\mathbf{h}} \right]_{i,j}
$$

$$
+ \left(\left[\Psi_{\mathrm{D}_1}^{\prime\,k,\mathbf{h}} \right]_{i,j} \left[\widetilde{J}_{1,22}^{k,\mathbf{h}} \right]_{i,j} + \gamma \left[\Psi_{\mathrm{D}_2}^{\prime\,k,\mathbf{h}} \right]_{i,j} \left[\widetilde{J}_{2,22}^{k,\mathbf{h}} \right]_{i,j} \right) \left[du_2^{k,\mathbf{h}} \right]_{i,j}
$$

$$
- \alpha \sum_{l \in \{x_1,x_2\}} \sum_{\tilde{i},\tilde{j} \in \mathcal{N}_l(i,j)} \frac{\left(\left[\Psi_{\mathrm{S}}^{\prime\,k,\mathbf{h}} \right]_{\tilde{i},\tilde{j}} + \left[\Psi_{\mathrm{S}}^{\prime\,k,\mathbf{h}} \right]_{i,j} \right)}{2} \frac{\left(\left[du_2^{k,\mathbf{h}} \right]_{\tilde{i},\tilde{j}} - \left[du_2^{k,\mathbf{h}} \right]_{i,j} \right)}{h_l^2}
$$

$$
+ \left(\left[\Psi_{\mathrm{D}_1}^{\prime\,k,\mathbf{h}} \right]_{i,j} \left[\widetilde{J}_{1,23}^{k,\mathbf{h}} \right]_{i,j} + \gamma \left[\Psi_{\mathrm{D}_2}^{\prime\,k,\mathbf{h}} \right]_{i,j} \left[\widetilde{J}_{2,23}^{k,\mathbf{h}} \right]_{i,j} \right)
$$

$$
- \alpha \sum_{l \in \{x_1,x_2\}} \sum_{\tilde{i},\tilde{j} \in \mathcal{N}_l(i,j)} \frac{\left(\left[\Psi_{\mathrm{S}}^{\prime\,k,\mathbf{h}} \right]_{\tilde{i},\tilde{j}} + \left[\Psi_{\mathrm{S}}^{\prime\,k,\mathbf{h}} \right]_{i,j} \right)}{2} \frac{\left(\left[u_2^{k,\mathbf{h}} \right]_{\tilde{i},\tilde{j}} - \left[u_2^{k,\mathbf{h}} \right]_{i,j} \right)}{h_l^2} = 0 \quad (5.29)
$$

for $i = 1, .., N_{x_1}^{k,\mathbf{h}}$ and $j = 1, .., N_{x_2}^{k,\mathbf{h}}$. Here, $\left[\Psi_{\mathrm{D}_1}^{\prime\,k,\mathbf{h}} \right]_{i,j}$, $\left[\Psi_{\mathrm{D}_2}^{\prime\,k,\mathbf{h}} \right]_{i,j}$ and $\left[\Psi_{\mathrm{S}}^{\prime\,k,\mathbf{h}} \right]_{i,j}$ denote the functions $\Psi_{\mathrm{D}_1}'(s^2), \Psi_{\mathrm{D}_2}'(s^2)$ and $\Psi_{\mathrm{S}}'(s^2)$ evaluated at pixel (i,j), i.e.

$$
\left[\Psi_{\mathrm{D}_1}^{\prime\,k,\mathbf{h}} \right]_{i,j} = \frac{1}{2\sqrt{\left(\left[du_1^{k,\mathbf{h}} \right]_{i,j}, \left[du_2^{k,\mathbf{h}} \right]_{i,j}, 1 \right)^\top (J_1^{k,\mathbf{h}})_{i,j} \left(\left[du_1^{k,\mathbf{h}} \right]_{i,j}, \left[du_2^{k,\mathbf{h}} \right]_{i,j}, 1 \right) + \epsilon_{\mathrm{D}_1}^2}},
$$

$$(5.30)$$

$$
\left[\Psi_{\mathrm{D}_2}^{\prime\,k,\mathbf{h}} \right]_{i,j} = \frac{1}{2\sqrt{\left(\left[du_1^{k,\mathbf{h}} \right]_{i,j}, \left[du_2^{k,\mathbf{h}} \right]_{i,j}, 1 \right)^\top (J_2^{k,\mathbf{h}})_{i,j} \left(\left[du_1^{k,\mathbf{h}} \right]_{i,j}, \left[du_2^{k,\mathbf{h}} \right]_{i,j}, 1 \right)) + \epsilon_{\mathrm{D}_2}^2}},
$$

$$(5.31)$$

$$
\left[\Psi_{\mathrm{S}}^{\prime\,k,\mathbf{h}} \right]_{i,j} = \frac{1}{2\sqrt{\left| D^{2,\mathbf{h}}\left(\left[u_1^{k,\mathbf{h}} \right]_{i,j} + \left[du_1^{k,\mathbf{h}} \right]_{i,j} \right) \right|^2 + \left| D^{2,\mathbf{h}}\left(\left[u_2^{k,\mathbf{h}} \right]_{i,j} + \left[du_2^{k,\mathbf{h}} \right]_{i,j} \right) \right|^2 + \epsilon_{\mathrm{S}}^2}},
$$

$$(5.32)$$

where $|D^{2,\mathbf{h}}\left(\left[z^{\mathbf{h}} \right]_{i,j} \right)|$ denotes the gradient magnitude operator as defined in Table 3.1. Please note that these abbreviations are still nonlinear expressions.

5.5.1.1 The Inexact Lagged-Diffusivity Method

Since we are interesting in developing a bidirectional multigrid method for solving the preceding nonlinear systems of equations, we need a suitable non-hierarchical iterative solver. Following our strategy from the nonlinear cases I and II, it makes sense to select an inexact lagged diffusivity method for this purpose (cf. Subsection 4.2.2).

Concrete Example. Let us show how such a solver can be realised for our prototype H. To this end, we consider a Gauß–Seidel method with frozen coefficient and coupled point relaxation (CPR). The corresponding iteration instructions for each equation system then read

$$
\begin{pmatrix} \left[du_1^{k,\mathbf{h}} \right]_{i,j}^{n+1} \\ \left[du_2^{k,\mathbf{h}} \right]_{i,j}^{n+1} \end{pmatrix} = \begin{pmatrix} \left[K_{11}^{k,\mathbf{h}} \right]_{i,j}^{n} & \left[K_{12}^{k,\mathbf{h}} \right]_{i,j}^{n} \\ \left[K_{12}^{k,\mathbf{h}} \right]_{i,j}^{n} & \left[K_{22}^{k,\mathbf{h}} \right]_{i,j}^{n} \end{pmatrix}^{-1} \begin{pmatrix} \left[b_1^{k,\mathbf{h}} \right]_{i,j}^{n} \\ \left[b_2^{k,\mathbf{h}} \right]_{i,j}^{n} \end{pmatrix} \quad (5.33)
$$

for $i = 1, .., N^{k,\mathbf{h}}_{x_1}$ and $j = 1, .., N^{k,\mathbf{h}}_{x_2}$ with the matrix entries

$$
\left[K^{k,\mathbf{h}}_{11}\right]^n_{i,j} = \left(\left[\Psi'^{k,\mathbf{h}}_{\mathrm{D}_1}\right]_{i,j}\left[\widetilde{J}^{k,\mathbf{h}}_{1,11}\right]_{i,j} + \gamma \left[\Psi'^{k,\mathbf{h}}_{\mathrm{D}_2}\right]_{i,j}\left[\widetilde{J}^{k,\mathbf{h}}_{2,11}\right]_{i,j}\right)
$$
$$
+\alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)} \frac{\left(\left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]_{\tilde{i},\tilde{j}} + \left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]_{i,j}\right)}{2h^2_l} \tag{5.34}
$$

$$
\left[K^{k,\mathbf{h}}_{12}\right]^n_{i,j} = \left(\left[\Psi'^{k,\mathbf{h}}_{\mathrm{D}_1}\right]_{i,j}\left[\widetilde{J}^{k,\mathbf{h}}_{1,12}\right]_{i,j} + \gamma \left[\Psi'^{k,\mathbf{h}}_{\mathrm{D}_2}\right]_{i,j}\left[\widetilde{J}^{k,\mathbf{h}}_{2,12}\right]_{i,j}\right) \tag{5.35}
$$

$$
\left[K^{k,\mathbf{h}}_{22}\right]^n_{i,j} = \left(\left[\Psi'^{k,\mathbf{h}}_{\mathrm{D}_1}\right]_{i,j}\left[\widetilde{J}^{k,\mathbf{h}}_{1,22}\right]_{i,j} + \gamma \left[\Psi'^{k,\mathbf{h}}_{\mathrm{D}_2}\right]_{i,j}\left[\widetilde{J}^{k,\mathbf{h}}_{2,22}\right]_{i,j}\right)
$$
$$
+\alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}(i,j)} \frac{\left(\left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]_{\tilde{i},\tilde{j}} + \left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]_{i,j}\right)}{2h^2_l} \tag{5.36}
$$

and right hand side

$$
\left[b^{k,\mathbf{h}}_1\right]^n_{i,j} = -\left(\left[\Psi'^{k,\mathbf{h}}_{\mathrm{D}_1}\right]_{i,j}\left[\widetilde{J}^{k,\mathbf{h}}_{1,13}\right]_{i,j} + \gamma \left[\Psi'^{k,\mathbf{h}}_{\mathrm{D}_2}\right]_{i,j}\left[\widetilde{J}^{k,\mathbf{h}}_{2,13}\right]_{i,j}\right)
$$
$$
+ \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}^-_l(i,j)} \frac{\left(\left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]^n_{\tilde{i},\tilde{j}} + \left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]^n_{i,j}\right)}{2h^2_l}\left[du^{k,\mathbf{h}}_1\right]^{n+1}_{\tilde{i},\tilde{j}}
$$
$$
+ \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}^+_l(i,j)} \frac{\left(\left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]^n_{\tilde{i},\tilde{j}} + \left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]^n_{i,j}\right)}{2h^2_l}\left[du^{k,\mathbf{h}}_1\right]^n_{\tilde{i},\tilde{j}}
$$
$$
+ \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)} \frac{\left(\left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]^n_{\tilde{i},\tilde{j}} + \left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]^n_{i,j}\right)}{2h^2_l}\left(\left[u^{k,\mathbf{h}}_1\right]_{\tilde{i},\tilde{j}} - \left[u^{k,\mathbf{h}}_1\right]_{i,j}\right)
$$
$$
\tag{5.37}
$$

$$
\left[b^{k,\mathbf{h}}_2\right]^n_{i,j} = -\left(\left[\Psi'^{k,\mathbf{h}}_{\mathrm{D}_1}\right]_{i,j}\left[\widetilde{J}^{k,\mathbf{h}}_{1,23}\right]_{i,j} + \gamma \left[\Psi'^{k,\mathbf{h}}_{\mathrm{D}_2}\right]_{i,j}\left[\widetilde{J}^{k,\mathbf{h}}_{2,23}\right]_{i,j}\right)
$$
$$
+ \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}^-_l(i,j)} \frac{\left(\left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]^n_{\tilde{i},\tilde{j}} + \left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]^n_{i,j}\right)}{2h^2_l}\left[du^{k,\mathbf{h}}_2\right]^{n+1}_{\tilde{i},\tilde{j}}
$$
$$
+ \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}^+_l(i,j)} \frac{\left(\left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]^n_{\tilde{i},\tilde{j}} + \left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]^n_{i,j}\right)}{2h^2_l}\left[du^{k,\mathbf{h}}_2\right]^n_{\tilde{i},\tilde{j}}
$$
$$
+ \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)} \frac{\left(\left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]^n_{\tilde{i},\tilde{j}} + \left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]^n_{i,j}\right)}{2h^2_l}\left(\left[u^{k,\mathbf{h}}_2\right]_{\tilde{i},\tilde{j}} - \left[u^{k,\mathbf{h}}_2\right]_{i,j}\right)
$$
$$
\tag{5.38}
$$

Please note that after each iteration the frozen coefficients, i.e. the nonlinear expressions (5.30)–(5.32) have to be updated.

5.5.1.2   Bidirectional Multigrid Methods

It is evident that the proposed frozen coefficient method in (5.33)–(5.33) suffers by construction from the same drawbacks as its counterpart for the nonlinear cases I and II: the weak attenuation of low error frequencies. As we have already seen in the context of the nonlinear cases I and II, so-called *full approximation schemes* (FAS) [Bra77] allow to overcome this attenuation problem that is common to all splitting-based iterative numerical solvers. By using residual equation systems to compute suitable correction steps on coarser grids, these schemes provide a useful framework for the efficient attenuation of all error frequencies. Thus, it is evidently desirable to develop such numerical methods also for the nonlinear case III. One should note that in contrast to all previous cases these schemes have now to be applied at *each* step/level of the fixed point iteration to solve the resulting nonlinear system of equation, i.e it they are used to solve the complete series of equation systems given by (5.28)– (5.29). To the best of our knowledge, this combination of a unidirectional optimisation scheme (coarse-to-fine incremental fixed point iteration) and a bidirectional numerical solver (full approximation scheme) has not been proposed in the literature so far.

Concrete Example. Let us now discuss how such a bidirectional numerical solver can be realised for our prototype H. Since the underlying strategy of a full approximation scheme has already been explained in Chapter 4, we restrict ourselves at this point to a brief discussion of the four required components:

I) *Basic Solver*. Firstly, we have to select a basic iterative solver. As indicated in Subsubsection 5.5.1.1 we propose the use of a Gauß-Seidel method with frozen coefficients and coupled point relaxation (CPR).

II) *Intergrid Transfer Operators*. Secondly, a suitable pair of intergrid transfer operators has to be defined. Since they have to fulfill the same requirements as in the nonlinear case I and II – the accuracy condition in (4.52) and the preservation of positive definiteness if applied channelwise to matrices – we choose once again area based interpolation and area based averaging [BWKS05].

III) *Coarse Grid Cell Size*. Thirdly, a new coarse grid cell size **H** has to be chosen. Also for all previous multigrid approaches, we decide for an *non-dyadic approach* with

$$H_{x_1} := h_{x_1} \frac{N_{x_1}^{k,\mathbf{h}}}{N_{x_1}^{k,\mathbf{H}}} \qquad\qquad H_{x_2} := h_{x_2} \frac{N_{x_2}^{k,\mathbf{h}}}{N_{x_2}^{k,\mathbf{H}}} \qquad\qquad (5.39)$$

with $N_{x_1}^{H} = \lceil N_{x_1}^{k,\mathbf{h}}/2 \rceil$ and $N_{x_2}^{k,\mathbf{H}} = \lceil N_{x_2}^{k,\mathbf{h}}/2 \rceil$.

IV) *Residual Coarse Grid Equation System*. Finally, we have to set up the the *nonlinear residual equation system* for the coarse grid. As we have already discussed in the nonlinear cases I and II, the discretisation coarse grid approximation approach (DCA) is a reasonable strategy to create a useful coarse grid representation of our nonlinear operator $A^{k,\mathbf{h}}(\mathbf{x}^{k,\mathbf{h}})$. Using this approach, we obtain the following coarse grid equation system:

$$\left(\left[\Psi'^{k,\mathbf{H}}_{\mathrm{D}_1}\right]_{i,j} \left[\widetilde{J}^{k,\mathbf{H}}_{1,11}\right]_{i,j} + \gamma \left[\Psi'^{k,\mathbf{H}}_{\mathrm{D}_2}\right]_{i,j} \left[\widetilde{J}^{k,\mathbf{H}}_{2,11}\right]_{i,j} \right) \left[du^{k,\mathbf{H}}_1\right]_{i,j} \tag{5.40}$$

$$+ \quad \left(\left[\Psi'^{k,\mathbf{H}}_{\mathrm{D}_1}\right]_{i,j} \left[\widetilde{J}^{k,\mathbf{H}}_{1,12}\right]_{i,j} + \gamma \left[\Psi'^{k,\mathbf{H}}_{\mathrm{D}_2}\right]_{i,j} \left[\widetilde{J}^{k,\mathbf{H}}_{2,12}\right]_{i,j} \right) \left[du^{k,\mathbf{H}}_2\right]_{i,j}$$

$$- \quad \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)} \frac{\left(\left[\Psi'^{k,\mathbf{H}}_{\mathrm{S}}\right]_{\tilde{i},\tilde{j}} + \left[\Psi'^{k,\mathbf{H}}_{\mathrm{S}}\right]_{i,j} \right)}{2} \frac{\left(\left[du^{k,\mathbf{H}}_1\right]_{\tilde{i},\tilde{j}} - \left[du^{k,\mathbf{H}}_1\right]_{i,j} \right)}{H^2_l}$$

$$+ \quad \left(\left[\Psi'^{k,\mathbf{H}}_{\mathrm{D}_1}\right]_{i,j} \left[\widetilde{J}^{k,\mathbf{H}}_{1,13}\right]_{i,j} + \gamma \left[\Psi'^{k,\mathbf{H}}_{\mathrm{D}_2}\right]_{i,j} \left[\widetilde{J}^{k,\mathbf{H}}_{2,13}\right]_{i,j} \right)$$

$$- \quad \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)} \frac{\left(\left[\Psi'^{k,\mathbf{H}}_{\mathrm{S}}\right]_{\tilde{i},\tilde{j}} + \left[\Psi'^{k,\mathbf{H}}_{\mathrm{S}}\right]_{i,j} \right)}{2} \frac{\left(\left[u^{k,\mathbf{H}}_1\right]_{\tilde{i},\tilde{j}} - \left[u^{k,\mathbf{H}}_1\right]_{i,j} \right)}{H^2_l} = \left[\tilde{f}^{k,\mathbf{H}}_1\right]_{i,j}$$

$$\left(\left[\Psi'^{k,\mathbf{H}}_{\mathrm{D}_1}\right]_{i,j} \left[\widetilde{J}^{k,\mathbf{H}}_{1,12}\right]_{i,j} + \gamma \left[\Psi'^{k,\mathbf{H}}_{\mathrm{D}_2}\right]_{i,j} \left[\widetilde{J}^{k,\mathbf{H}}_{2,12}\right]_{i,j} \right) \left[du^{k,\mathbf{H}}_1\right]_{i,j} \tag{5.41}$$

$$+ \quad \left(\left[\Psi'^{k,\mathbf{H}}_{\mathrm{D}_1}\right]_{i,j} \left[\widetilde{J}^{k,\mathbf{H}}_{1,22}\right]_{i,j} + \gamma \left[\Psi'^{k,\mathbf{H}}_{\mathrm{D}_2}\right]_{i,j} \left[\widetilde{J}^{k,\mathbf{H}}_{2,22}\right]_{i,j} \right) \left[du^{k,\mathbf{H}}_2\right]_{i,j}$$

$$- \quad \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)} \frac{\left(\left[\Psi'^{k,\mathbf{H}}_{\mathrm{S}}\right]_{\tilde{i},\tilde{j}} + \left[\Psi'^{k,\mathbf{H}}_{\mathrm{S}}\right]_{i,j} \right)}{2} \frac{\left(\left[du^{k,\mathbf{H}}_2\right]_{\tilde{i},\tilde{j}} - \left[du^{k,\mathbf{H}}_2\right]_{i,j} \right)}{H^2_l}$$

$$+ \quad \left(\left[\Psi'^{k,\mathbf{H}}_{\mathrm{D}_1}\right]_{i,j} \left[\widetilde{J}^{k,\mathbf{H}}_{1,23}\right]_{i,j} + \gamma \left[\Psi'^{k,\mathbf{H}}_{\mathrm{D}_2}\right]_{i,j} \left[\widetilde{J}^{k,\mathbf{H}}_{2,23}\right]_{i,j} \right)$$

$$- \quad \alpha \sum_{l\in\{x_1,x_2\}} \sum_{\tilde{i},\tilde{j}\in\mathcal{N}_l(i,j)} \frac{\left(\left[\Psi'^{k,\mathbf{H}}_{\mathrm{S}}\right]_{\tilde{i},\tilde{j}} + \left[\Psi'^{k,\mathbf{H}}_{\mathrm{S}}\right]_{i,j} \right)}{2} \frac{\left(\left[u^{k,\mathbf{H}}_2\right]_{\tilde{i},\tilde{j}} - \left[u^{k,\mathbf{H}}_2\right]_{i,j} \right)}{H^2_l} = \left[\tilde{f}^{k,\mathbf{H}}_1\right]_{i,j}$$

for $i = 1, ..., N^{k,\mathbf{H}}_{x_1}$ and $j = 1, ..., N^{k,\mathbf{H}}_{x_2}$, where $\left[du^{k,\mathbf{H}}_1\right]_{i,j}$ and $\left[du^{k,\mathbf{H}}_2\right]_{i,j}$ are the new unknowns and $\left[\tilde{f}^{k,\mathbf{H}}_1\right]_{i,j}$ and $\left[\tilde{f}^{k,\mathbf{H}}_2\right]_{i,j}$ are the modified right hand side (cf. (4.92)). Thereby the entries of the coarse grid motion tensors $\left[\widetilde{J}^{k,\mathbf{H}}_{1,12}\right]_{i,j}$ and $\left[\widetilde{J}^{k,\mathbf{H}}_{2,12}\right]_{i,j}$ are obtained via channelwise restriction, while the different nonlinear expressions $\left[\Psi'^{k,\mathbf{H}}_{\mathrm{D}_1}\right]_{i,j}$, $\left[\Psi'^{k,\mathbf{H}}_{\mathrm{D}_2}\right]_{i,j}$ and $\left[\Psi'^{k,\mathbf{H}}_{\mathrm{S}}\right]_{i,j}$ have to be adapted with respect to the new coarse grid cell size $\mathbf{H}$:

$$\left[\Psi'^{k,\mathbf{H}}_{\mathrm{D}_1}\right]_{i,j} = \frac{1}{2\sqrt{\left(\left[du^{k,\mathbf{H}}_1\right]_{i,j}, \left[du^{k,\mathbf{H}}_2\right]_{i,j}, 1 \right)^\top (J^{k,\mathbf{H}}_1)_{i,j} \left(\left[du^{k,\mathbf{H}}_1\right]_{i,j}, \left[du^{k,\mathbf{H}}_2\right]_{i,j}, 1 \right) + \epsilon^2_{\mathrm{D}_1}}} ,$$

$$\tag{5.42}$$

$$\left[\Psi'^{k,\mathbf{H}}_{\mathrm{D}_2}\right]_{i,j} = \frac{1}{2\sqrt{\left(\left[du^{k,\mathbf{H}}_1\right]_{i,j}, \left[du^{k,\mathbf{H}}_2\right]_{i,j}, 1 \right)^\top (J^{k,\mathbf{H}}_2)_{i,j} \left(\left[du^{k,\mathbf{H}}_1\right]_{i,j}, \left[du^{k,\mathbf{H}}_2\right]_{i,j}, 1 \right) + \epsilon^2_{\mathrm{D}_2}}} ,$$

$$\tag{5.43}$$

$$\left[\Psi'^{k,\mathbf{H}}_{\mathrm{S}}\right]_{i,j} = \frac{1}{2\sqrt{\left| D^{2,\mathbf{H}}\left(\left[u^{k,\mathbf{H}}_1\right]_{i,j} + \left[du^{k,\mathbf{H}}_1\right]_{i,j} \right) \right|^2 + \left| D^{2,\mathbf{H}}\left(\left[u^{k,\mathbf{H}}_2\right]_{i,j} + \left[du^{k,\mathbf{H}}_2\right]_{i,j} \right) \right|^2 + \epsilon^2_{\mathrm{S}}}} ,$$

$$\tag{5.44}$$

As in the nonlinear cases I and II, the adaptation of $\left[\Psi'^{\mathbf{h}}_{\mathrm{S}}\right]_{i,j}$ results from the adaptation of the diffusion tensors $\left[T^{k,\mathbf{h}}\right]_{i,j} = \left[\Psi'^{k,\mathbf{h}}_{\mathrm{S}}\right]_{i,j} I$.

In order to speed up the computation even further we implemented a full multigrid variant of our full approximation scheme. This in turn means that we are actually combining three different multigrid ideas: (i) the incremental coarse-to-fine fixed point iteration to avoid local minima, (ii) at each step of this iteration a unidirectional multigrid method for a hierarchical initialisation and (iii) at each level of this unidirectional multigrid method a bidirectional multigrid method as solver (for attenuating all error frequencies efficiently).

This example shows that our systematic framework for the design of efficient multigrid schemes from Chapter 4 can also be extended to variational methods for large displacements. To this end, it has to be combined with an incremental coarse-to-fine fixed point iteration – a unidirectional multigrid strategy that avoids local minima during the minimisation. The obtained numerical method offers two advantages: (i) It allows to maintain the systematic and compact notation based on motion and diffusion tensors. (ii) It permits to adapt all numerical concepts that we have proposed in the context of the nonlinear cases I and II. This in turn means that we have actually established a complete framework for the design of efficient numerical schemes for variational optic flow methods in general.

5.5.2   Numerical Prototypes

As in the case of small displacements, we are interested in evaluating the efficiency of different numerical methods for our prototype H. To this end, we have selected the same numerical techniques as in our comparison in Section 4.3: Apart from the proposed full approximation scheme (see Table 5.9), we considered one unidirectional multigrid method (also based on our multigrid framework), one basic and one advanced non-hierarchical solver (one Gauß-Seidel and one SOR based technique), and two techniques that make use of the elliptic minimisation strategy (one explicit and one semi-implicit approach; the latter also based on SOR). In order to allow for a fair comparison, all solvers have been embedded in the same incremental coarse-to-fine fixed point iteration as our full approximation scheme.

Tab. 5.9: Implemented bidirectional multigrid schemes for the different variational models. MG = multigrid. WARP = incremental coarse-to-fine fixed point iteration (warping). FMG = full multigrid. W = W-cycle. Cyc = multigrid cycles per level. GS = Gauß–Seidel. CPR = coupled point relaxation. ALR = alternating line relaxation. Pre/Post = pre- and postsmoothing relaxation iterations. L = linear. NL = nonlinear.

Case	Model	MG Solver	Cyc	Basic Solver	Pre/Post
L	A - Homogeneous	FMG-W	1	GS-CPR	1-1
	B - Image-Driven Isotropic	FMG-W	2	GS-CPR	2-2
	C - Image-Driven Anisotropic	FMG-W	4	GS-ALR	1-1 -
NL I	D - Flow-Driven Isotropic	FAS-FMG-W	2	GS-CPR	2-2
	E - Flow-Driven Anisotropic	FAS-FMG-W	4	GS-ALR	1-1
NL II	F - Bruhn *et al.* 2-D, SD	FAS-FMG-W	2	GS-CPR	2-2
	G - Papenberg *et al.* 3-D, SD	FAS-FMG-W	2	GS-CPR	2-2
NL III	**H - Bruhn/Weickert 2-D, LD**	**WARP-FAS-FMG-W**	**2**	**GS-CPR**	**3-3**

5.6 Numerical Experiments

After we have introduced our numerical prototypes let us start with our experimental evaluation. Also this time, all computations are carried out on a standard desktop PC with a 3.06 GHz Intel Pentium 4 CPU executing C / C++ code.

5.6.1 Prototypes for Large Displacements

In order to evaluate the efficiency of the different numerical schemes for our prototype H, we selected a downsampled variant (160×120) of the *Rheinhafen* sequence. As in the case of small displacements, a relative error of $e_{rel} = 10^{-2}$ was used as stopping criterion. However, one should note that this relative error now not only refers to a single nonlinear equation system. It refers to a whole series of systems that results from the proposed hierarchical minimisation via the incremental coarse-to-fine fixed point iteration. This in turn means that errors on coarser levels influence the results on finer levels such that errors can propagate.

The Nonlinear Case III. Let us now take a look at the obtained results in Table 5.10. They clearly show that even in the case of such highly accurate optic flow methods as our prototype H, frame rates can be achieved that are close to real-time (3 frames per second for $e_{rel} = 10^{-2}$; up to 7 frames per second with slightly reduced accuracy requirements [BW05]). Since apart from the incremental coarse-to-fine fixed point iteration the same numerical schemes are used as for the nonlinear case II, it is not surprising that the corresponding speedups are similar to the ones obtained for our advanced prototypes F and G (cf. Section 4.4). As one can see, also this time our full multigrid implementation is the most efficient solver: While it outperforms the basic Gauß-Seidel method and its unidirectional multigrid variant by two orders of magnitude, it is even almost four orders of magnitude more efficient than the stabilised explicit scheme. With respect to both techniques based on the SOR method – the inexact Lagged-Diffusivity solver and the

Tab. 5.10: Performance benchmark for the 2-D variant of the method of Bruhn/Weickert [BW05]. The benchmark was performed on a standard desktop computer with 3.06 GHz Pentium 4 CPU. Run times refer to the computation of a single flow field from the downsampled *Rheinhafen* sequence (size 160×120). FPS = frames per second.

<div align="center">

Nonlinear Case III: Prototype H (Bruhn/Weickert 2-D)
Separately robustified data term with additional gradient constancy (without linearisation)
+ flow-driven isotropic regularisation
$\sigma = 0.0, \alpha = 300, \gamma = 10, \eta = 0.50, \epsilon_{D_1} = 10^{-1}, \epsilon_{D_2} = 10^{-1}, \epsilon_S = 10^{-3}$

</div>

Solver at *each* warping level	Iterations	Time $[s]$	FPS $[s^{-1}]$	Speedup
Mod. Explicit Scheme ($\tau = 0.00025$)	174217	1906.546	0.001	1
Gauß-Seidel (CPR)	9798	74.816	0.013	25
Cascadic Gauß-Seidel (CPR)	1646	16.190	0.062	118
Semi-Implicit Scheme ($\tau = 10000$) + SOR	25/5	0.664	1.506	2868
Incomplete Lagged SOR ($\omega = 1.98$)	25/5	0.617	1.621	3089
FAS - Full Multigrid	**1**	**0.349**	**2.861**	**5454**

semi-implicit scheme – we can observe a speedup of a factor two. This is also accordance with our benchmarks for the nonlinear case II. Even the superiority of the elliptic strategy becomes evident one more time: The semi-implicit scheme performs best if the time step size τ is chosen very large. This in turn means that it approximates de facto the inexact Lagged-Diffusivity method, i.e. the parabolic strategy comes down to the elliptic one.

Quality. In Figure 5.11 the computed flow field for our prototype H is presented and compared to the one of our prototype G. Evidently, both flow fields looks fairly realistic. However, one can clearly see the advantages of our prototype H by looking at two details: (i) the decreasing displacements of the van in the foreground with increasing distance to the camera and (ii) cars at left and right side of the background that are not visible in the estimation of our prototype G. Please note that this highly accurate flow field is just one example of the quality of the proposed near real-time implementation. Of course, all results presented in our qualitative experiments for large displacements can be reproduced with our full multigrid method.
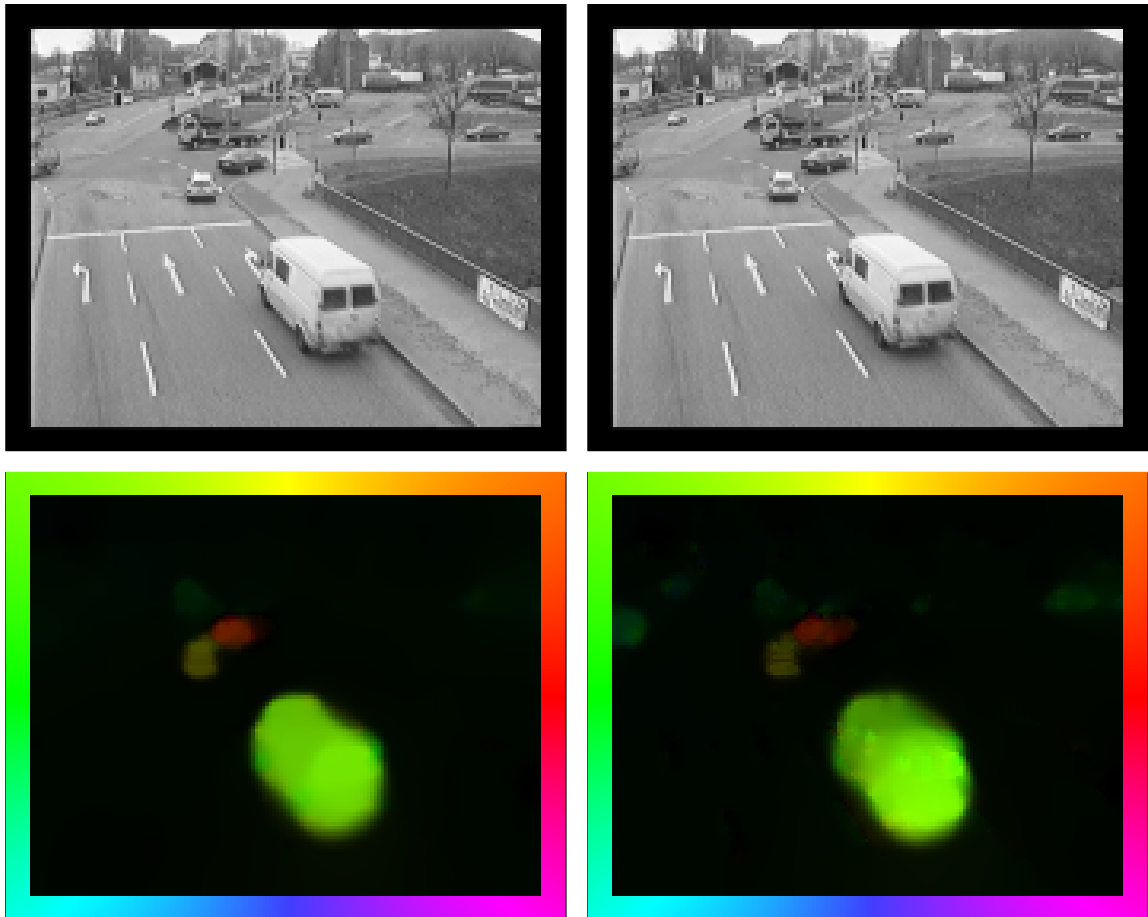

Fig. 5.11: (a) **Top Left:** Frame 1130 of the *Rheinhafen* sequence by Nagel (resized to 160×120). (b) **Top Right:** Frame 1131. (c) **Bottom Left:** Computed flow field by our full multigrid implementation for the prototype G (Papenberg *et al.* 3-D, SD). Computing time: 121 milliseconds. (d) **Bottom Right:** Computed flow field by our full multigrid implementation for the prototype H (Bruhn/Weickert 2-D, LD). Computing time: 349 milliseconds.

5.7 Summary

In this chapter we completed our work on the accurate modelling and the efficient minimisation of variational optic flow methods by discussing theoretical and numerical aspects of approaches that are suitable for *large displacements*.

Starting point of our efforts was the introduction of a *novel variational approach* that served as eighth prototype for our qualitative and numerical studies: the high accuracy method of Bruhn and Weickert. By this example we demonstrated that the problem of large displacements can be tackled, if the linearisation of the constancy assumptions is *postponed* from the modelling phase to the optimisation scheme. Then, instead of compromising the continuous model and therewith the estimation quality, the linearisation may even be helpful to speed up the computation.

In this context, we also gave a theoretical justification of the warping technique that was motivated so far only on an algorithmic basis: We showed that this method can be derived as an incremental coarse-to-fine fixed point iteration that is employed to minimise energy functionals of variational methods based on constancy assumptions *without* linearisation. This finding also allowed us to extend our framework based on motion and diffusion tensors to the case of large displacements.

In the experimental evaluation of the estimation quality, all these efforts were rewarded by excellent results: Our prototype not only proved to be more accurate than our previous methods with respect to the estimation of large displacements, it even yielded the currently most accurate results of all methods presented in the literature so far. This demonstrated once more that performance and deeper theoretical understanding are not contradictive: They are nothing else but two sides of the same medal.

Finally, we also extended our framework for the design of efficient multigrid methods to the case of large displacements. Thereby we exploited the fact that the proposed incremental coarse-to-fine fixed point iteration required the solution of nonlinear systems of equations that have the same structure as those in Chapter 4. This in turn allowed us to reuse all numerical concepts for nonlinear methods that we had already discussed so that the development of a suitable bidirectional multigrid method for our new prototype became straightforward.

As in the case of small displacements, our efficiency benchmarks showed the superiority of the presented framework: With speedups of two to four orders of magnitude compared to standard basic solvers and unidirectional multigrid schemes, our implementation yielded once more the best results. Thereby real-time near frame rates of up to three frames per second for sequences of size 160×120 demonstrated that there is no reason to refrain from accurate optic flow techniques if a fast performance is required: If highly efficient numerical schemes are used one can have both advantages at the same time.

6

Summary and Outlook

6.1 Summary

The main topic of this thesis was to introduce a systematic approach for the accurate design and the efficient implementation of variational optic flow methods. This was done successively in four steps:

Accurate Modelling for Small Displacements

In Chapter 2 we investigated the modelling of convex variational optic flow methods for small displacements. To this end, we reviewed various established concepts for designing both the data and the smoothness term. Apart from discussing existing concepts we also introduced *several novel ideas* that addressed important problems in the field of motion estimation such as varying illumination, non-motion-invariant image features and corrupted data due to noise and outliers. Moreover, we presented a *systematic notation* for variational optic flow methods based on motion and diffusion tensors. This notation offered us three advantages: (i) It allowed for a compact representation of the essential properties of the underlying optic flow methods, (ii) it simplified the construction and implementation of novel approaches in Chapter 3, and (iii) it was the key to a successful design of efficient numerical schemes (as shown in Chapter 4 and 5). Thus, together with our presentation and evaluation of the different concepts, we provided a *general toolkit* for the systematic construction of variational optic flow methods for small displacements.

Design and Comparison of Suitable Prototypes

In Chapter 3 we made use of our toolkit from Chapter 2 and presented seven variational approaches that served as prototypes for our qualitative and numerical studies. To this end, we considered one approach for each of the five regularisation strategies proposed in [WS01a] – among them the popular methods of Horn and Schunck [HS81] and Nagel and Enkelmann [NE86]. Moreover, we developed *two novel techniques* as prototypes for the class of advanced variational methods: the noise robust approach of Bruhn *et al.* and the illumination robust optic flow method of Papenberg *et al.*. Comparisons to state-of-the-art methods from the literature showed that these advanced prototypes not only yield excellent results, they even *competed successfully* with methods that were explicitly constructed for large displacements. In order to compute the unique minimiser of all seven energy functionals we proposed to follow an elliptic strategy and solve the associated Euler–Lagrange equations. As a useful byproduct of the motion and diffusion tensor notation, we thereby obtained a *taxonomy* for variational optic flow approaches. This taxonomy allowed us to focus our numerical efforts on three representative classes of methods: approaches with (i) linear terms only, (ii) linear and nonlinear terms, and (iii) nonlinear terms only.

Design and Comparison of Efficient Numerical Schemes

In Chapter 4 we investigated efficient numerical methods for the solution of this three different classes of linear and nonlinear equations systems. Our main focus was thereby the development of bidirectional multigrid schemes that are among the fastest numerical methods for fulfilling this task. In this context we not only derived efficient numerical schemes for all our prototypes, we even succeeded to establish a *complete multigrid framework* for variational optic flow methods. This framework that is based on the diffusion and motion tensor notation as well as the discretisation coarse grid approximation approach (DCA) [BHM00] allowed the systematic construction of efficient hierarchical solvers for all variational methods that were designed in accordance with our systematic toolkit from Chapter 2. Performance benchmarks for image sequences of size 160×120 showed the efficiency of the developed hierarchical methods: With frame rates of up to 63 frames per second, they allowed for the *first time in the literature* to compute variational optic flow in *real-time*. These frame rates correspond to significant speedups of two to four order of magnitude compared to standard iterative solvers such as the Gauß–Seidel method and stabilised explicit schemes.

Extension to Large Displacements

In chapter 5 we extended our framework for the accurate modelling and the efficient minimisation of variational optic flow methods to the case of large displacements. To this end, we lifted the constraint with respect to the strict design of convex energy functionals and introduced a *third novel (nonconvex) approach*: the high accuracy method of Bruhn and Weickert. Comparisons to state-of-the-art methods from the literature showed the excellent performance of this prototype: It yielded results that were *more accurate than those of any other technique* presented in the literature so far. Main reasons for this performance was our strategy to postpone the linearisation of all constancy assumptions to the optimisation scheme and to use an incremental coarse-to-fine fixed point iteration for the minimisation of the resulting nonconvex energy functional. In this context, we also succeeded to *justify the well-known warping method theoretically* that was only motivated on an experimental basis so far. We demonstrated that it can be naturally derived as a numerical approximation strategy for solving the Euler–Lagrange equations of energy functionals with constancy assumptions in their original nonlinear from.

Apart from designing the currently most precise optic flow method, we also showed how to extended our motion and diffusion tensor notation to the case of large displacements. Thereby we saw that each of the nonlinear equation systems that is obtained by discretising the incremental coarse-to-fine fixed iterations can be efficiently solved by means of the same multigrid framework that we have established in Chapter 4. Performance benchmarks for our prototype for sequences sequences of size 160×120 showed the same efficiency of our multigrid schemes as in the case of small displacements: With up to 3 frames per second *near real-time* frame rates were achieved for such high quality optic flow methods for the first time in the literature. Thereby once more significant speedups of two to four order of magnitude compared to standard iterative solvers were achieved.

With the work that was done in this thesis we have demonstrated that *high quality optic flow computation and real-time performance are not opposing worlds*: They can be combined if accurate modelling and efficient numerics go hand in hand.

6.2   Future Work

Although the proposed framework allows the construction of very fast and highly accurate optic flow techniques, there still remains a lot of things that can be done. In the following we will sketch some of the most promising ideas:

Modelling Ideas

Let us start with three aspects that concern an improved modelling or the integration of additional concepts in the existing framework.

- *Occlusions.* One important aspect that was excluded from our modelling framework from Chapter 2 is the *correct handling of occlusions*. In this context one may think of a symmetrical approach that determines the forward and backward displacement field at the same time. In [ADPS02] such a method is presented: It couples the estimation in both directions by an additional terms that penalises deviations from bijectivity. If such a hard constraint makes sense or not is not clear. However, methods that integrate the occlusion handling directly in the estimation process are most likely to give better result than approaches that postpone this task to a post-processing step. Therefore, the investigation of suitable integration of the occlusion handling is one of the most important tasks.

- *Problem Specific Regularisers.* Another interesting aspect that exceeds our current modelling framework is the use of approaches that include physics in the regulariser. In particular if the expected type of motion is non-rigid, such types of approaches may be very useful. Typical fields of applications are the estimation of motion in fluid dynamics [CMP02] and meteorology [KS03] as well as the registration of medical images [Mod04]. In this context, one should note that such non-rigid regularisers may also lead to an explicit point coupling via the smoothness term. This in turn requires a non-trivial extension of our taxonomy for Euler–Lagrange equations in Chapter 4 and the development of modified numerical schemes.

- *Motion Segmentation*. A last interesting idea is the *combination of segmentation and optic flow*. In this context level set methods have become more and more popular [CS05, AK05a, AK05b]. The main idea is thereby to determine the number of regions and the corresponding flow fields at the same time. Apart from the perfect preservation of motion discontinuities at regions boundaries, such a proceeding also allows for a compact representation of the image sequence. This in turn can be exploited by subsequent tasks such as video compression and video indexing. A first own approach that gives impressive results with respect to standard test sequences will be presented in [BBW06]. The development of efficient numerical solvers for such combined techniques, however, is still an open task.

Numerical Ideas

Apart from the development of suitable bidirectional multigrid methods for the previous three modelling concepts, there are also three general numerical ideas that may be worth being investigated.

- *Iterant Recombination.* Firstly, in order to improve the performance of our bidirectional multigrid method even further, one may think of *recombining results from previous iterations* [BM95]. In particular, if the representation of the coarse grid operator is poor, such a strategy is known to be useful. Promising results in the context of variational optic flow methods have been recently presented in [KCR05]. However, these results were restricted so simple techniques with image-driven regularisation (linear case).

- *Multilevel Nonlinear Methods.* Secondly, one may consider the implementation of multilevel nonlinear methods (MNM). Such nonlinear multigrid techniques that have been recently proposed in [YD06] combine the advantages of full approximation schemes (FAS) that are based on nonlinear correction steps (c.f. Chapter 4) and Newton-Rhapson methods that perform linear updates. Thereby the nonlinear equation system is split into a strongly and a weakly nonlinear part. Since each of the parts are processed by the correspondingbasic method – FAS for the nonlinear and Newton-Rhapson for the linear part – almost all nonlinear equation systems can be solved more efficiently.

- *Direct Parallelisation.* Thirdly, one may think of a *direct parallelisation* of the proposed multigrid schemes. In contrast to domain decomposition techniques that tackle the problem indirectly by decomposing the problem [KSBW04, KSBW05], such a direct strategy only distributes the computation itself and thus requires a much lesser computational overhead. This in turn allows for higher speedups with similar numbers of CPUs. Recent results for a direct parallelisation of the method of Horn and Schunck presented in [KCR05] confirm the usefulness of the latter approach.

Extension to Other Correspondence Problems

Motion estimation is not the only important correspondence problem in computer vision. Stereo reconstruction and medical image registration are other widely researched representatives of this type of problems. Evidently, it seems desirable to extend our modelling and multigrid framework also to these fields of application. However, at least in the case of stereo reconstruction this is a non-trivial task.

- *Stereo Reconstruction.* The accurate estimation of displacement fields from stereo camera systems requires the integration of the epipolar constraint [Fau93]. Since this constraint limits the search space of the underlying correspondence problem to certain lines (epipolar lines), the number of unknowns is reduced from two to one per pixels. However, in return, the structure of the equations gets significantly more complex. Although we already presented a first successful approach in [SBW05], the design of suitable methods is still not straightforward: Both the modelling and the numerical implementation require significant adaptations.

All these ideas give just a small impression on the realm of things that can still be done. However, one thing is for sure: *There will always be a need for fast and accurate algorithms in computer vision.*

A   Notation

f	grey value image or image sequence
$\mathbf{f} = (f_{(1)}, ..., f_{(m)})^\top$	multi-channel image or image sequence
x_1, x_2	spatial coordinates in an image or image sequence
x_3	temporal Coordinate in an image sequence
$\mathbf{x}$	coordinate vector, i.e., either $\mathbf{x} = (x_1, x_2)$ in images or $\mathbf{x}^\top = (x_1, x_2, x_3)^\top$ in image sequences
t	diffusion time in diffusion–reaction systems
Ω_2, Ω_3	rectangular image domain, $\Omega_2 \subset \mathbb{R}^2$ for images, $\Omega_3 \subset \mathbb{R}^2 \times \mathbb{R}_0^+$ for image sequences
$|\Omega|$	size of the image domain Ω
∂_a	abbreviation for $\frac{\partial}{\partial a}$
a_b	abbreviation for $\frac{\partial a}{\partial b}$
a_{bc}	abbreviation for $\frac{\partial^2 a}{\partial b \partial c}$
$\nabla_2 a$	spatial gradient of a, i.e., $(\partial_{x_1} a, \partial_{x_2} a)^\top$
$\nabla_3 a$	spatiotemporal gradient of a, i.e., $(\partial_{x_1} a, \partial_{x_2} a, \partial_{x_3} a)^\top$
$\Delta_2 a$	spatial Laplacian of a, i.e., $\partial_{x_1}^2 a + \partial_{x_2}^2 a$
$\Delta_3 a$	spatiotemporal Laplacian of a, i.e., $\partial_{x_1}^2 a + \partial_{x_2}^2 a + \partial_{x_3}^2 a$
$\mathcal{H}_2 a$	spatial Hessian of a, i.e., $(\partial_{x_1}^2 a, \partial_{x_1} \partial_{x_2} a)^\top, (\partial_{x_1} \partial_{x_2} a, \partial_{x_2}^2 a)^\top)$
$\mathrm{div}(\mathbf{a})$	$\partial_{x_1} a_1 + \partial_{x_2} a_2$ in the spatial case and $\partial_{x_1} a_1 + \partial_{x_2} a_2 + \partial_{x_3} a_3$ in the spatiotemporal case
$|\mathbf{a}|$	magnitude of vector $\mathbf{a}$
u_1, u_2	optic flow component in x_1 and x_2 direction
$\mathbf{u} = (u_1, u_2, 1)^\top$	optic flow vector, see also next entry
$\mathbf{u} = (u_1, ..., u_m)^\top$	evolving image in the case of vector-valued diffusion filtering

$D^k a$ set of derivatives of a with order k

$M(D^k f, \mathbf{u})$ data term

$S(\nabla f, \nabla \mathbf{u})$ smoothness term

α weight of the smoothness term

γ_i weight of constancy assumption i

μ_i weight of sequence channel i

K_σ gaussian kernel with standard deviation σ

σ_{spat} spatial standard deviation of Gaussian kernel

σ_{temp} temporal standard deviation of Gaussian kernel

$K_\sigma * f$ convolution of f with K_σ

σ_n standard deviation of Gaussian noise

J motion tensor: 3×3 matrix

T diffusion tensor: either 2×2 matrix in the spatial or 3×3 matrix in the spatiotemporal case

$\lambda_1, ..., \lambda_n$ eigenvalues of a $n \times n$ matrix

$\mathbf{v}_1, ..., \mathbf{v}_n$ eigenvectors of a $n \times n$ matrix

$\mathrm{rank}\,(A)$ rank of matrix A

A_{ij} entry of matrix A at position (i, j)

$A(t)$ matrix processed by nonlinear diffusion after time t

$g(s^2)$ diffusivity

$\epsilon_{\mathrm{D}}, \epsilon_{\mathrm{S}}$ small regularisation parameter for the data and the smoothness term

$\Psi_{\mathrm{D}}(s^2), \Psi_{\mathrm{S}}(s^2)$ nonquadratic penaliser of the data and the smoothness term

$\Psi'_{\mathrm{D}}(s^2), \Psi'_{\mathrm{S}}(s^2)$ derivatives of $\Psi_{\mathrm{D}}(s^2)$ and $\Psi_{\mathrm{S}}(s^2)$ with respect to s^2

$\mathcal{L}_{\mathrm{L}}$ linear differential operator

$\mathcal{L}_{\mathrm{NL}}$ nonlinear differential operator

h_l grid cell size in direction of dimension l

$\mathbf{h}$ grid cell size vector $\mathbf{h} = (h_{x_1}, h_{x_2})^\top$ in the spatial case and $\mathbf{h} = (h_{x_1}, h_{x_2}, h_{x_3})^\top$ in the spatiotemporal case

$N_l^{\mathbf{h}}$	number of grid cells in direction of dimension l if discretised with grid cell size $\mathbf{h}$
$N^{\mathbf{h}}$	total number of grid cells if discretised with grid cell size $\mathbf{h}$
$a^{\mathbf{h}}, \mathbf{a}^{\mathbf{h}}, A^{\mathbf{h}}$	expression a, $\mathbf{a}$ and A discretised with grid cell size $\mathbf{h}$
$\left[a^{\mathbf{h}}\right]_{i,j}, \left[\mathbf{a}^{\mathbf{h}}\right]_{i,j}, \left[A^{\mathbf{h}}\right]_{i,j}$	expression a, $\mathbf{a}$ and A for cell (i,j) discretised with grid cell size $\mathbf{h}$
$\left[\Psi_{\mathrm{D}}'^{\mathbf{h}}\right]_{i,j}$	evaluated expression of $\Psi_{\mathrm{D}}'(s^2)$ at pixel (i,j)
$\left[\Psi_{\mathrm{S}}'^{\mathbf{h}}\right]_{i,j}$	evaluated expression of $\Psi_{\mathrm{S}}'(s^2)$ at pixel (i,j)
$\mathrm{diag}(\mathbf{a}^{\mathbf{h},\top})$	diagonal matrix with entries of vector $\mathbf{a}^{\mathbf{h}}$ as main diagonal
$L_{\mathrm{L}}^{\mathbf{h}}$	discrete linear differential operator
$L_{\mathrm{NL}}^{\mathbf{h}}$	discrete nonlinear differential operator
$F_{\mathrm{NL}}^{\mathbf{h}}$	discrete nonlinear operator derived from nonquadratic data term
$A^{\mathbf{h}}$	linear operator (system matrix)
$A^{\mathbf{h}}(\mathbf{x}^{\mathbf{h}})$	nonlinear operator
$\mathbf{x}^{\mathbf{h}}$	vector of unknowns
$\mathbf{b}^{\mathbf{h}}$	right hand side
$D^{\mathbf{h}}$	diagonal part of a matrix
$U^{\mathbf{h}}$	strictly upper diagonal of a matrix
$L^{\mathbf{h}}$	strictly lower diagonal of a matrix
$\mathcal{N}(i,j)$	set of neighbours of cell (i,j)
$|\mathcal{N}(i,j)|$	number of neighbours of cell (i,j)
$\mathcal{N}_l(i,j)$	set of neighbours of cell (i,j) in direction of dimension l
$\mathcal{N}_l^{-}(i,j)$	set of neighbours of cell (i,j) in direction of dimension l that have still to be processed
$\mathcal{N}_l^{+}(i,j)$	set of neighbours of cell (i,j) in direction of dimension l that have already been processed
$\mathcal{B}$	set of cells (used in block relaxation algorithm)
$\mathbf{x}_{\mathcal{B}_i}^{\mathbf{h}}$	vector containing only those entries of $\mathbf{x}^{\mathbf{h}}$ that are in the set $\mathcal{B}_i$
$A_{\mathcal{B}_i \mathcal{B}_j}^{\mathbf{h}}$	matrix with those entries of $A^{\mathbf{h}}$ that relate the unknowns in the sets B_i and B_j

ω	overrelaxation parameter for the SOR method
H_l	coarse grid cell size in direction of dimension l
$\mathbf{H}$	coarse grid cell size vector $\mathbf{H} = (H_{x_1}, H_{x_2})^\top$ in the spatial case and $\mathbf{H} = (H_{x_1}, H_{x_2}, H_{x_3})^\top$ in the spatiotemporal case
$P^{\mathbf{H} \to \mathbf{h}}$	prolongation operator
$R^{\mathbf{h} \to \mathbf{H}}$	restriction operator
$\lceil a \rceil$	smallest integer number b with $b \geq a$
$\tilde{x}^{\mathbf{h}}$	approximative solution of $\mathbf{x}^{\mathbf{h}}$
$\mathbf{r}^{\mathbf{h}}$	residual $\mathbf{b}^{\mathbf{h}} - A^{\mathbf{h}} \tilde{\mathbf{x}}^{\mathbf{h}}$
$\mathbf{e}^{\mathbf{h}}$	error $\mathbf{x}^{\mathbf{h}} - \tilde{\mathbf{x}}^{\mathbf{h}}$
ν_1	number of presmoothing iterations (bidirectional multigrid)
ν_2	number of postsmoothing iterations (bidirectional multigrid)
du_1^l, du_2^l	motion increment in x_1 and x_2 direction for fixed point step l
$\widetilde{M}(D^k f, \mathbf{u})$	data term based on constancy assumptions without linaerisatio
$\widetilde{J}$	motion tensor for large displacements: 3×3 matrix
η	downsampling factor in the interval $(0, 1)$
τ	time step size in parabolic approaches

B

<div style="text-align: right">**Own Publications**</div>

Book Chapters

1. P. Mrázek, J. Weickert, A. Bruhn. On robust estimation and smoothing with spatial and tonal kernels. In R. Klette, R. Kozera, L. Noakes, J. Weickert (Eds.): *Geometric Properties from Incomplete Data.* Springer, Dordrecht, 335–352, 2006.

2. A. Bruhn, J. Weickert. Confidence measures for variational optic flow methods. In R. Klette, R. Kozera, L. Noakes, J. Weickert (Eds.): *Geometric Properties from Incomplete Data.* Springer, Dordrecht, 283–297, 2006.

3. J. Weickert, A. Bruhn, T. Brox, and N. Papenberg. A survey on variational methods for small displacements, O. Scherzer (Ed.), *Mathematical Models for Registration and Applications to Medical Imaging*, Mathematics in Industry, Vol. 10, Springer, Berlin, 2006, to appear.

Journal Papers

4. A. Bruhn, T. Jakob, M. Fischer, T. Kohlberger, J. Weickert, U. Brüning, and C. Schnörr. High performance cluster computing with 3-D nonlinear diffusion filters. *Real-Time Imaging*, Vol. 10, No. 1, 41–51, 2004.

5. A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, Vol. 61, No. 3, 1–21, Febr./March 2005.

6. A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnörr. Variational optic flow computation in real-time. *IEEE Transactions in Image Processing*, Vol. 14, No. 5, 608–615, May 2005.

7. T. Kohlberger, C. Schnörr, A. Bruhn, and J. Weickert. Domain decomposition for variational optical flow computation. *IEEE Transactions in Image Processing*, Vol. 14, No. 8, 1125–1137, August 2005.

8. N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, Special Issue – European Conference on Computer Vision 2004, 2006, to appear. **Invited Paper**.

9. B. Burgeth, N. Papenberg, A. Bruhn, M. Welk, and J. Weickert. Mathematical Morphology for Tensor Data Induced by the Loewner Ordering in Higher Dimensions. *Signal Processing*, Special Issue – Tensor Signal Processing, 2006, to appear.

10. B. Burgeth, A. Bruhn, S. Didas, J. Weickert, and M. Welk. Morphology for Tensor Data: Ordering versus PDE-Based Approach. *Image and Vision Computing*, Special Issue – International Symposium on Mathematical Morphology 2005, 2006, to appear. **Invited Paper**.

11. A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr. Discontinuity-preserving variational optic flow computation in real-time. In *International Journal of Computer Vision*, Special Issue – International Conference on Scale Space and PDE Methods in Computer Vision 2005, 2006, to appear. **Invited Paper**.

Conference Papers

12. A. Bruhn, T. Jakob, M. Fischer, T. Kohlberger, J. Weickert, U. Brüning, and C. Schnörr. Designing 3-D nonlinear diffusion filters for high performance cluster computing, L. Van Gool (Ed.), *Pattern Recognition*, Lecture Notes in Computer Science, Vol. 2449, Springer, Berlin, 290–297, 2002.

13. A. Bruhn, J. Weickert, and C. Schnörr. Combining the advantages of local and global optic flow methods, L. Van Gool (Ed.), *Pattern Recognition*, Lecture Notes in Computer Science, Vol. 2449, Springer, Berlin, 454–462, 2002. **Awarded a DAGM 2002 Paper Prize.**

14. A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnörr. Real-time optic flow computation with variational methods. In N. Petkov, M. A. Westenberg (Eds.): *Computer Analysis of Images and Patterns*. Lecture Notes in Computer Science, Vol. 2756, Springer, Berlin, 222–229, 2003.

15. T. Kohlberger, C. Schnörr, A. Bruhn, and J. Weickert. Domain decomposition for parallel variational optic flow computation. In B. Michaelis, G. Krell (Eds.): *Pattern Recognition*. Lecture Notes in Computer Science, Vol. 2781, Springer, Berlin, 196–202, 2003.

16. D. Slogsnat, M. Fischer, A. Bruhn, J. Weickert, and U. Brüning. Low level parallelization of nonlinear diffusion filtering algorithms for cluster computing environments. In H. Kosch, L. Böszörményi, H. Hellwagner (Eds.): *Euro-Par 2003. Parallel Processing*. Lecture Notes in Computer Science, Vol. 2790, Springer, Berlin, 481–490, 2003.

17. N. Papenberg, A. Bruhn, T. Brox, and J. Weickert. Numerical justification for multiresolution optical flow computation. In L. Alvarez (Ed.): IWCVIA '03: *International Workshop on Computer Vision and Image Analysis*. Vol. 0026 von Cuardernos del Instituto Universitario de Ciencias y Technologias Ciberneticas, University Las Palmas de Gran Canaria, Spain, 7–12, 2004.

18. J. Weickert, A. Bruhn, N. Papenberg, and T. Brox. Variational optic flow computation: From continuous models to algorithms. *International Workshop on Computer Vision and Image Analysis*. Vol. 0026 von Cuardernos del Instituto Universitario de Ciencias y Technologias Ciberneticas, University Las Palmas de Gran Canaria, Spain, 1–6, 2004.

19. T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In T. Pajdla, J. Matas (Eds.): *Computer Vision - ECCV 2004*. Lecture Notes in Computer Science, Vol. 3024, Springer, Berlin, 25–36, 2004. **Received The Longuet-Higgins Best Paper Award.**

20. T. Kohlberger, C. Schnörr, A. Bruhn, and J. Weickert. Parallel variational motion estimation by domain decomposition and cluster computing. In T. Pajdla, J. Matas (Eds.): *Computer Vision - ECCV 2004*. Lecture Notes in Computer Science, Vol. 3024, Springer, Berlin, 205–216, 2004.

21. B. Burgeth, N. Papenberg, A. Bruhn, M. Welk, C. Feddern, and J. Weickert. Mathematical morphology based on the Loewner ordering for tensor data. In *Proc. 2005 International Symposium on Mathematical Morphology*, Springer, Berlin, 407–418, 2005.

22. A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr. Discontinuity-preserving computation of variational optic flow in real-time. In R. Kimmel, N. Sochen, J. Weickert (Eds.): *Scale-Space and PDE Methods in Computer Vision*. Lecture Notes in Computer Science, Vol. 3459, Springer, Berlin, 279–290, 2005.

23. N. Slesareva, A. Bruhn, and J. Weickert. Optic flow goes stereo: A variational method for estimating discontinuity-preserving dense disparity maps. In W. Kropatsch, R. Sablatnig, A. Hanbury (Eds.): *Pattern Recognition*, Lecture Notes in Computer Science, Vol. 3663, Springer, Berlin, 33–40, 2005. **Awarded a DAGM 2005 Paper Prize.**

24. A. Bruhn, and J. Weickert. Towards ultimate motion estimation: Combining highest accuracy and real-time performance, In *Proc. 10th International Conference on Computer Vision*, Beijing, China. IEEE Computer Society Press, Vol.1, 749–755, 2005.

25. I. Galić, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, and H.-P. Seidel. Towards PDE-based image compression, In *Proc. 3rd Workshop on Variational, Geometric and Level Set Methods in Computer Vision*. Lecture Notes in Computer Science, Vol. 3459, Springer, Berlin, 37–48, 2005.

26. T. Brox, A. Bruhn, and J. Weickert. Variational motion segmentation with level sets. In H. Bischof, A. Leonardis, A. Pinz (Eds.): *Computer Vision - ECCV 2006*. Lecture Notes in Computer Science, Springer, Berlin, to appear.

Bibliography

[ADK99]   G. Aubert, R. Deriche, and P. Kornprobst. Computing optical flow via variational techniques. *SIAM Journal on Applied Mathematics*, 60(1):156–182, 1999.

[ADPS02]   L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sánchez. Symmetrical dense optical flow estimation with occlusions detection. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Computer Vision – ECCV 2002*, volume 2350 of *Lecture Notes in Computer Science*, pages 721–736. Springer, Berlin, 2002.

[ADSW02]   L. Alvarez, R. Deriche, J. Sánchez, and J. Weickert. Dense disparity map estimation respecting image derivatives: a PDE and scale-space based approach. *Journal of Visual Communication and Image Representation*, 13(1/2):3–21, 2002.

[AELS99]   L. Alvarez, J. Esclarín, M. Lefébure, and J. Sánchez. A PDE model for computing the optical flow. In *Proc. XVI Congreso de Ecuaciones Diferenciales y Aplicaciones*, pages 1349–1356, Las Palmas de Gran Canaria, Spain, September 1999.

[AK02]   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, New York, 2002.

[AK05a]   T. Amiaz and N. Kiryati. Dense discontinuous optical flow via contour-based segmentation. In *Proc. Twelfth International Conference on Image Processing*, volume 3, pages 1264–1267, Genoa, Italy, September 2005.

[AK05b]   T. Amiaz and N. Kiryati. Piecewise-smooth dense optical flow via level sets. Technical Report VIA-2005-6-2, Vision and Image Analysis Laboratory, School of Electrical Engineering, Tel Aviv University, Israel, June 2005.

[Ana89]   P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[AWS99]   L. Alvarez, J. Weickert, and J. Sánchez. A scale-space approach to nonlocal optical flow calculations. In M. Nielsen, P. Johansen, O. F. Olsen, and J. Weickert, editors, *Scale-Space Theories in Computer Vision*, volume 1682 of *Lecture Notes in Computer Science*, pages 235–246. Springer, Berlin, 1999.

[AWS00]   L. Alvarez, J. Weickert, and J. Sánchez. Reliable estimation of dense optical flow fields with large displacements. *International Journal of Computer Vision*, 39(1):41–56, August 2000.

[Axe97]   O. Axelsson. Globally convergent continuation methods for non-linear equations. In *Proc. First Workshop on Large-Scale Scientific Computations*, International House of Scientists "F. Curie", Bulgaria, 1997.

[BA91]   M. J. Black and P. Anandan. Robust dynamic motion estimation over time. In *Proc. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 292–302, Maui, HI, June 1991. IEEE Computer Society Press.

[BA96]   M. J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, January 1996.

[BAHH92]   J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In F. Hodnett, editor, *Proc. Sixth European Conference on Mathematics in Industry*, pages 237–252. Teubner, Stuttgart, 1992.

[BAK91]   R. Battiti, E. Amaldi, and C. Koch. Computing optical flow across multiple scales: An adaptive corse-to-fine strategey. *International Journal of Computer Vision*, 6(2):133–145, 1991.

[BB84]   B. F. Buxton and H. Buxton. Computation of optic flow from the motion of edges in image sequences. *Image and Vision Computing*, 2:59–75, 1984.

[BBPW04]   T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optic flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *Computer Vision – ECCV 2004*, volume 3024 of *Lecture Notes in Computer Science*, pages 25–36. Springer, Berlin, 2004.

[BBW06]   T. Brox, A. Bruhn, and J. Weickert. Variational motion segmentation with level sets. In H. Bischof, A. Leonardis, and A. Pinz, editors, *Computer Vision – ECCV 2006*, Lecture Notes in Computer Science. Springer, Berlin, 2006. To appear.

[BC98]   P. Blomgren and T. F. Chan. Color TV: total variation methods for restoration of vector valued images. *IEEE Transactions on Image Processing*, 7(3):304–309, March 1998.

[BD96]   F.A. Bornemann and P. Deuflhard. The cascadic multigrid method for elliptic problems. *Numerische Mathematik*, 75:135–152, 1996.

[BFB94]   J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, February 1994.

[BGW91]   J. Bigün, G. H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, August 1991.

[BHM00]   W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. SIAM, Philadelphia, second edition, 2000.

[BHS98]   A. Bab-Hadiashar and D. Suter. Robust optic flow computation. *International Journal of Computer Vision*, 29(1):59–77, August 1998.

[BIK02]   A. Borzi, K. Ito, and K. Kunisch. Optimal control formulation for determining optical flow. *SIAM Journal on Scientific Computing*, 24(3):818–847, 2002.

[BJ96]   M. J. Black and A. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):972–986, October 1996.

[Bla94]   M. J. Black. Recursive non-linear estimation of discontinuous flow fields. In J.-O. Eklundh, editor, *Computer Vision – ECCV '94*, volume 800 of *Lecture Notes in Computer Science*, pages 138–145. Springer, Berlin, 1994.

[BM95]   A. Brandt and V. Mikulinsky. On recombining iterants in multigrid algorithms and problems with small islands. *SIAM Journal on Scientific Computing*, 16:20–28, 1995.

[BPT88]   M. Bertero, T. A. Poggio, and V. Torre. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889, August 1988.

[Bra77]   A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, April 1977.

[Bro05]   T. Brox. *From pixels to regions: partial differential equations in image analysis*. PhD thesis, Saarland University, Saarbrücken, Germany, 2005.

[Bru01]   A. Bruhn. Regularization in motion estimation. Master's thesis, Department of Mathematics and Computer Science, University of Mannheim, Germany, October 2001.

[BvdBL+06]   T. Brox, R. van den Boomgaard, F. Lauze, J. van de Weijer, J. Weickert, P. Mrázek, and P. Kornprobst. Adaptive structure tensors and their applications. In J. Weickert and H. Hagen, editors, *Visualization and Processing of Tensor Fields*, pages 17–47, Berlin, January 2006. Springer.

[BW02]   T. Brox and J. Weickert. Nonlinear matrix diffusion for optic flow estimation. In L. Van Gool, editor, *Pattern Recognition*, volume 2449 of *Lecture Notes in Computer Science*, pages 446–453. Springer, Berlin, 2002.

[BW05]   A. Bruhn and J. Weickert. Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In *Proc. Tenth International Conference on Computer Vision*, volume 1, pages 749–755, Beijing, China, June 2005. IEEE Computer Society Press.

[BWF⁺03]   A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnörr. Real-time optic flow computation with variational methods. In N. Petkov and M. A. Westberg, editors, *Computer Analysis of Images and Patterns*, volume 2756 of *Lecture Notes in Computer Science*, pages 222–229. Springer, Berlin, 2003.

[BWF⁺05]   A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnörr. Variational optical flow computation in real-time. *IEEE Transactions on Image Processing*, 14(5):608–615, May 2005.

[BWKS05]   A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr. Discontinuity-preserving computation of variational optic flow in real-time. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale-Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*, pages 279–290. Springer, Berlin, 2005.

[BWKS06]   A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr. A multigrid platform for real-time motion computation with discontinuity-preserving variational methods. *International Journal of Computer Vision*, 2006. To appear.

[BWS02]   A. Bruhn, J. Weickert, and C. Schnörr. Combining the advantages of local and global optic flow methods. In L. Van Gool, editor, *Pattern Recognition*, volume 2449 of *Lecture Notes in Computer Science*, pages 454–462. Springer, Berlin, 2002.

[BWS05]   A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[BYJ97]   M. J. Black, Y. Yacoob, and X. S. Ju. Recognizing human motion using parameterized models of optical flow. In M. Shah and R. Jain, editors, *Motion-Based Recognotion*, pages 245–269. Kluwer, Boston, 1997.

[CBFAB94]   P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proc. 1994 IEEE International Conference on Image Processing*, volume 2, pages 168–172, Austin, TX, November 1994. IEEE Computer Society Press.

[CCW97]   R. H. Chan, T. F. Chan, and W. L. Wan. Multigrid for differential-convolution problems arising from image processing. In G. Golub, S. H. Lui, F. Luk, and R. Plemmons, editors, *Proc. Workshop on Scientific Computing*, pages 58–72, Hong Kong, September 1997.

[CH53]   R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 1. Interscience, New York, 1953.

[CM99]   T. F. Chan and P. Mulet. On the convergence of the lagged diffusivity fixed point method in total variation image restoration. *SIAM Journal on Numerical Analysis*, 36(2):354–367, 1999.

[CMP02]   T. Corpetti, E. Mémin, and P. Pérez. Dense estimation of fluid flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):365–380, 2002.

[Coh93]   I. Cohen. Nonlinear variational method for optical flow computation. In *Proc. Eighth Scandinavian Conference on Image Analysis*, volume 1, pages 523–530, Tromsø, Norway, May 1993.

[CS05]   D. Cremers and S. Soatto. Motion competition: A variational framework for piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265, May 2005.

[DK02]   G. N. DeSouza and A. C. Kak. Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, 2002.

[DKA95]   R. Deriche, P. Kornprobst, and G. Aubert. Optical-flow estimation while preserving its discontinuities: a variational approach. In *Proc. Second Asian Conference on Computer Vision*, volume 2, pages 290–295, Singapore, December 1995.

[Els61]   L. E. Elsgolc. *Calculus of Variations*. Pergamon, Oxford, 1961.

[Enk87]   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, 1987.

[Enk91]   W. Enkelmann. Obstacle detection by evaluation of optical flow fields from image sequences. *Image and Vision Computing*, 9(3):160–168, 1991.

[ER03]   M. El Kalmoun and U. Rüde. A variational multigrid for computing the optical flow. In T. Ertl, B. Girod, G. Greiner, H. Niemann, H.-P. Seidel, E. Steinbach, and R. Westermann, editors, *Vision, Modelling and Visualization*, pages 577–584. IOS Press, 2003.

[Far00]   G. Farnebäck. Fast and accurate motion estimation using orientation tensors and parametric motion models. In *Proc. 15th International Conference on Pattern Recognition*, volume 1, pages 135–139, Barcelona, Spain, September 2000.

[Far01]   G. Farnebäck. Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field. In *Proc. Eighth International Conference on Computer Vision*, volume 1, pages 171–177, Vancouver, Canada, July 2001. IEEE Computer Society Press.

[Fau93]   O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA, 1993.

[Fel05]   M. Felsberg. Optical flow estimation from monogenic phase. In *1st International Workshop on Complex Motion (IWCM)*, volume 3417 of *Lecture Notes in Computer Science*, Günzburg, Tyskland, October 2005. In press.

[FG87]   W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, Interlaken, Switzerland, June 1987.

[FJ90]   D. J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104, August 1990.

[FKN73]   S. Fučik, A. Kratochvil, and J. Nečas. Kačanov–Galerkin method. *Commentationes Mathematicae Universitatis Carolinae*, 14(4):651–659, 1973.

[For05]   H. Foroosh. Pixelwise-adaptive blind optic flow assuming nonstationary statistics. *IEEE Transactions on Image Processing*, 14(2):222–230, February 2005.

[Fri92]   D. S. Fritsch. A medial description of greyscale image structure by gradient-limited diffusion. In R. A. Robb, editor, *Visualization in Biomedical Computing '92*, volume 1808 of *Proceedings of SPIE*, pages 105–117. SPIE Press, Bellingham, 1992.

[FSHW04]   C. Frohn-Schnauf, S. Henn, and K. Witsch. Nonlinear multigrid methods for total variation denosing. *Computating and Visualization in Sciene*, 7(3–4):199–206, 2004.

[GF00]   I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Dover, New York, 2000.

[GKKJ92]   G. Gerig, O. Kübler, R. Kikinis, and F. A. Jolesz. Nonlinear anisotropic filtering of MRI data. *IEEE Transactions on Medical Imaging*, 11:221–232, 1992.

[Gla84]   F. Glazer. Multilevel relaxation in low-level computer vision. In A. Rosenfeld, editor, *Multiresolution Image Processing and Analysis*, pages 312–330. Springer, Berlin, 1984.

[GMN+98]   B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: an analysis of eight optical flow algorithms. In *Proc. 1998 British Machine Vision Conference*, Southampton, England, September 1998.

[GvV96]   S. Ghosal and P. Č. Vaněk. Scalable algorithm for discontinuous optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):181–194, February 1996.

[Hac85]   W. Hackbusch. *Multigrid Methods and Applications*. Springer, New York, 1985.

[HB93]   F. Heitz and P. Bouthemy. Multimodal estimation of discontinuous optical flow using Markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1217–1232, December 1993.

[Hei02]   S. Heinrich. Real time fusion of motion and stereo using flow/depth constraint for fast obstacle detection. In L. Van Gool, editor, *Pattern Recognition*, volume 2449 of *Lecture Notes in Computer Science*, pages 75–82. Springer, Berlin, 2002.

[HJ94]   R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, UK, 1994.

[HP01]   S. C. Han and C. I. Podilchuk. Video compression with dense motion fields. *IEEE Transactions on Image Processing*, 10(11):1605–1612, November 2001.

[HRRS86]   F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. MIT Press, Cambridge, MA, 1986.

[HS81]   B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[HSSW02]   W. Hinterberger, O. Scherzer, C. Schnörr, and J. Weickert. Analysis of optical flow models in the framework of calculus of variations. *Numerical Functional Analysis and Optimization*, 23(1/2):69–89, May 2002.

[Hub81]   P. J. Huber. *Robust Statistics*. Wiley, New York, 1981.

[Iij59]   T. Iijima. Basic theory of pattern observation. In *Papers of Technical Group on Automata and Automatic Control*. IECE, Japan, December 1959. In Japanese.

[Iij62]   T. Iijima. Observation theory of two-dimensional visual patterns. In *Papers of Technical Group on Automata and Automatic Control*. IECE, Japan, October 1962. In Japanese.

[Jäh93]   B. Jähne. *Spatio-Temporal Image Processing*, volume 751 of *Lecture Notes in Computer Science*. Springer, Berlin, 1993.

[JBJ96]   S. Ju, M. Black, and A. Jepson. Skin and bones: multi-layer, locally affine, optical flow and regularization with transparency. In *Proc. 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 307–314, San Francisco, CA, June 1996. IEEE Computer Society Press.

[KCR05]   H. Köstler, I. Christadler, and U. Rüde. Robust and efficient multigrid for the optic flow problem using different regularisers. Technical report, Institut für Informatik, Friedrich Alexander Universität Erlangen-Nürnberg, Germany, 2005.

[KDTN92]   D. Koller, K. Daniilidis, T. Thórhallson, and H.-H. Nagel. Model-based object tracking in traffic scenes. In Giulio Sandini, editor, *Computer Vision – ECCV '92*, volume 588 of *Lecture Notes in Computer Science*, pages 437–452. Springer, Berlin, 1992.

[KMS00]   R. Kimmel, R. Malladi, and N. Sochen. Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images. *International Journal of Computer Vision*, 39(2):111–129, September 2000.

[KN90]   D. Koller and H.-H. Nagel. Ein robustes verfahren zur detektion und verfolgung bewegter objekte in bildfolgen. In R. E. Großkopf, editor, *DAGM-Symposium*, volume 254 of *Informatik-Fachberichte*, pages 625–633, Oberkochen-Aalen, Germany, 1990.

[KNPS68]   J. Kačur, J. Nečas, J. Polák, and J. Souček. Convergence of a method for solving the magnetostatic field in nonlinear media. *Aplikace Matematiky*, 13:456–465, 1968.

[KS03]   T. Kohlberger and C. Schnörr. Variational dense motion estimation using the helmholtz decomposition. In L. D. Griffin and M. Lillholm, editors, *Scale-Space Theory in Computer Vision*, volume 2695 of *Lecture Notes in Computer Science*, pages 432–448, Berlin, 2003. Springer.

[KSBW04]   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, *Computer Vision – ECCV 2004*, volume 3024 of *Lecture Notes in Computer Science*, pages 205–216. Springer, Berlin, 2004.

[KSBW05]   T. Kohlberger, C. Schnörr, A. Bruhn, and J. Weickert. Domain decomposition for variational optical flow computation. *IEEE Transactions on Image Processing*, 14(8):1125–1137, 2005.

[KTB96]   A. Kumar, A. R. Tannenbaum, and G. J. Balas. Optic flow: a curve evolution approach. *IEEE Transactions on Image Processing*, 5(4):598–610, April 1996.

[KY03]   A. Kimmel and I. Yavneh. An algebraic multigrid approach for image analysis. *SIAM Journal on Scientific Computing*, 24(4):1218–1231, 2003.

[LC01]   M. Lefébure and L. D. Cohen. Image registration, optical flow and local rigidity. *Journal of Mathematical Imaging and Vision*, 14(2):131–147, March 2001.

[LCR98]   H. Liu, R. Chellappa, and A. Rosenfeld. Accurate dense optical flow estimation using adaptive structure tensors and a parametric model. *IEEE Transactions on Image Processing*, 7(3):292–303, March 1998.

[LK81]   B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Seventh International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, Canada, August 1981.

[LKW94]   M. R. Luettgen, W. C. Karl, and A. S. Willsky. Efficient multiscale regularization with applications to the computation of optical flow. *IEEE Transactions on Image Processing*, 3(1):41–64, 1994.

[LMB02]   G. Lefaix, Éric Marchand, and Patrick Bouthemy. Motion-based obstacle detection and tracking for card driving assistance. In *Proc. 16th International Conference on Pattern Recognition*, volume 4, pages 74–77, Quebec City, Canada, August 2002.

[Luc84]   B. D. Lucas. *Generalized Image Matching by the Method of Differences*. PhD thesis, School of Computer Science, Carnegie–Mellon University, Pittsburgh, PA, 1984.

[LV98]   S.-H. Lai and B. C. Vemuri. Reliable and efficient computation of optical flow. *International Journal of Computer Vision*, 29(2):87–105, October 1998.

[Mar]   J. M. Martínez. *MPEG-7 Overview*. International Organisation for Standardisation, Palma de Mallorca, Spain. http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm.

[MB87]   D. W. Murray and B. F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):220–228, March 1987.

[Mei99]   A. Meister. *Numerik linearer Gleichungssysteme*. Vieweg, Braunschweig, 1999.

[Mid03]   M. Middendorf. *Zur Auswertung lokaler Grauwertstrukturen*. PhD thesis, Department of Computer Science, University of Karlsruhe, Germany, July 2003.

[MKW97]   P. Moulin, R. Krishnamurthy, and J. W. Woods. Multiscale modeling and estimation of motion fields for video coding. *IEEE Transactions on Image Processing*, 6(12):1606–1620, December 1997.

[Mod04]   J. Modersitzki. *Numerical Methods for Image Registration*. Oxford University Press, 2004.

[MP98a]   E. Mémin and P. Pérez. Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE Transactions on Image Processing*, 7(5):703–719, May 1998.

[MP98b]   E. Mémin and P. Pérez. A multigrid approach for hierarchical motion estimation. In *Proc. 6th International Conference on Computer Vision*, pages 933–938, Bombay, India, January 1998.

[MP02]   E. Mémin and P. Pérez. Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision*, 46(2):129–155, 2002.

[Nag83a]   H.-H. Nagel. Constraints for the estimation of displacement vector fields from image sequences. In *Proc. Eighth International Joint Conference on Artificial Intelligence*, volume 2, pages 945–951, Karlsruhe, West Germany, August 1983.

[Nag83b]   H.-H. Nagel. Displacements vectors derived from second-order intensity variations in image sequences. *Computer Vision, Graphics and Image Processing*, 21:85–117, 1983.

[Nag90]   H.-H. Nagel. Extending the 'oriented smoothness constraint' into the temporal domain and the estimation of derivatives of optical flow. In O. Faugeras, editor, *Computer Vision – ECCV '90*, volume 427 of *Lecture Notes in Computer Science*, pages 139–148. Springer, Berlin, 1990.

[NE86]   H.-H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:565–593, 1986.

[Nes93]   P. Nesi. Variational approach to optical flow estimation managing discontinuities. *Image and Vision Computing*, 11(7):419–439, September 1993.

[NG98]   H.-H. Nagel and A. Gehrke. Spatiotemporally adaptive estimation and segmentation of OF-fields. In H. Burkhardt and B. Neumann, editors, *Computer Vision – ECCV '98*, volume 1407 of *Lecture Notes in Computer Science*, pages 86–102. Springer, Berlin, 1998.

[NH98]   H.-H. Nagel and M. Haag. Bias-corrected optical flow estimation for road vehicle tracking. In *Proc. Sixth International Conference on Computer Vision*, pages 1006–1011, Bombay, India, January 1998.

[NS98]   M. Z. Nashed and O. Scherzer. Least squares and bounded variation regularization with nondifferentiable functionals. *Numerical Functional Analysis and Optimization*, 19:873–901, 1998.

[ON95]   M. Otte and H.-H. Nagel. Estimation of optical flow based on higher-order spatiotemporal derivatives in interlaced and non-interlaced image sequences. *Artificial Intelligence*, 78:5–43, 1995.

[OR00]   J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*, volume 30 of *Classics in Applied Mathematics*. SIAM, Philadelphia, 2000.

[PBB+06]   N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 2006. To appear.

[PBBW03]   N. Papenberg, A. Bruhn, T. Brox, and J. Weickert. Numerical justification for multiresolution optical flow computation. In L. Alvárez, editor, *Proc. International Workshop on Computer Vision and Image Analysis*, volume 26 of *Cuadernos del Instituto Universitario de Ciencias y Tecnologias Ciberneticas*, pages 7–12, Las Palmas, Spain, September 2003.

[PVPO94]   M. Proesmans, L. Van Gool, E. Pauwels, and A. Oosterlinck. Determination of optical flow and its discontinuities using non-linear diffusion. In J.-O. Eklundh, editor, *Computer Vision – ECCV '94*, volume 801 of *Lecture Notes in Computer Science*, pages 295–304. Springer, Berlin, 1994.

[RB05]   S. Roth and M. Black. On the spatial statistices of optical flow. In *Proc. Tenth International Conference on Computer Vision*, volume 1, pages 42–49, Beijing, China, June 2005. IEEE Computer Society Press.

[RD96]   L. Robert and R. Deriche. Dense depth map reconstruction: A minimization and regulariza-
            tion approach which preserves discontinuities. In B. Buxton and R. Cipolla, editors, *Computer
            Vision – ECCV '96*, volume 1064 of *Lecture Notes in Computer Science*, pages 439–451.
            Springer, Berlin, 1996.

[ROF92]   L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms.
            Physica D, 60:259–268, 1992.

[RS91]   A. R. Rao and B. G. Schunck. Computing oriented texture fields. *CVGIP: Graphical Models
            and Image Processing*, 53:157–185, 1991.

[SAH91]   E. P. Simoncelli, E. H. Adelson, and D. J. Heeger. Probability distributions of optical flow. In
            Proc. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,
            pages 310–315, Maui, HI, June 1991. IEEE Computer Society Press.

[Sap01]   G. Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge Univer-
            sity Press, Cambridge, UK, 2001.

[SB02]   J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer, New York, third
            edition, 2002.

[SBW05]   N. Slesareva, A. Bruhn, and J. Weickert. Optic flow goes stereo: a variational approch for
            estimating discontinuity-preserving dense dispartiy maps. In W. Kropatsch, R. Sablatnig,
            and A. Hanbury, editors, *Pattern Recognition*, volume 3663 of *Lecture Notes in Computer
            Science*, pages 33–40. Springer, Berlin, 2005.

[SC94]   R. Szeliski and J. Coughlan. Hierarchical spline-based image registration. In *Proc. 1994 IEEE
            Computer Society Conference on Computer Vision and Pattern Recognition*, pages 194–201,
            Seattle, WA, June 1994. IEEE Computer Society Press.

[Sch93]   C. Schnörr. On functionals with greyvalue-controlled smoothness terms for determining op-
            tical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:1074–1079,
            1993.

[Sch94a]   C. Schnörr. Bewegungssegmentation von Bildfolgen durch die Minimierung konvexer nicht-
            quadratischer Funktionale. In W. Kropatsch and H. Bischof, editors, *Mustererkennung 1994*,
            pages 178–185. Springer, Berlin, 1994.

[Sch94b]   C. Schnörr. Segmentation of visual motion by minimizing convex non-quadratic functionals.
            In *Proc. Twelfth International Conference on Pattern Recognition*, volume A, pages 661–663,
            Jerusalem, Israel, October 1994. IEEE Computer Society Press.

[Sch97]   H. R. Schwarz. *Numerische Mathematik*. Teubner, Stuttgart, fourth edition, 1997.

[SH89]   D. Shulman and J. Hervé. Regularization of discontinuous flow fields. In *Proc. Workshop on
            Visual Motion*, pages 81–90, Irvine, CA, March 1989. IEEE Computer Society Press.

[Sin90]   A. Singh. An estimation-theoretic framework for image-flow computation. In *Proc. Third
            International Conference on Computer Vision*, pages 168–177. IEEE Computer Society Press,
            1990.

[TA77]   A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill–Posed Problems*. Wiley, Washington, DC,
            1977.

[TD01]   D. Tschumperlé and R. Deriche. Diffusion tensor regularization with contraints preservation.
            In *Proc. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recogni-
            tion*, volume 1, pages 948–953, Kauai, HI, December 2001. IEEE Computer Society Press.

[Ter86]   D. Terzopoulos. Image analysis using multigrid relaxation. *IEEE Transactions on Pattern
            Analysis and Machine Intelligence*, 8(2):129–139, March 1986.

[Tho49]   L. H. Thomas. Elliptic problems in linear difference equations over a network. Technical
            report, Watson Scientific Computing Laboratory, Columbia University, New York, NJ, 1949.

[Tis94]   M. Tistarelli. Multiple constraints for optical flow. In J.-O. Eklundh, editor, *Computer Vision – ECCV '94*, volume 801 of *Lecture Notes in Computer Science*, pages 61–70. Springer, Berlin, 1994.

[TLCH05]   C.-H. Teng, S.-H. Lai, Y.-S. Chen, and W.-H. Hsu. Accurate optical flow computation under non-uniform brightness variations. *Computer Vision and Image Understanding*, 97:315–346, 2005.

[TOS01]   U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, San Diego, 2001.

[TP84]   O. Tretiak and L. Pastor. Velocity estimation from image sequences with second order differential operators. In *Proc. Seventh International Conference on Pattern Recognition*, pages 16–19, Montreal, Canada, July 1984.

[UGVT88]   S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biological Cybernetics*, 60:79–87, 1988.

[Var00]   R. A. Varga. *Matrix Iterative Analysis*. Springer, New York, second edition, 2000.

[Vog95]   C. R. Vogel. A multigrid method for total variation-based image denosing. *Computation and Control IV*, 20:323–331, 1995.

[Vog02]   C. R. Vogel. *Computational Methods for Inverse Problems*. SIAM, Philadelphia, 2002.

[WAB06]   J. Wills, S. Agarwal, and S. Belongie. A feature-based approach for dense segmentation and estimation of large disparity motion. *International Journal of Computer Vision*, 2006. To appear.

[WaJCL98]   Y.-T. Wu, T. Kanade an J. Cohn, and C.-C. Li. Optical flow estimation using wavelet motion model. In *Proc. Sixth International Conference on Computer Vision*, pages 992–998, Bombay, India, January 1998. Narosa Publishing House.

[WB02]   J. Weickert and T. Brox. Diffusion and regularization of vector- and matrix-valued images. In M. Z. Nashed and O. Scherzer, editors, *Inverse Problems, Image Analysis, and Medical Imaging*, volume 313 of *Contemporary Mathematics*, pages 251–268. AMS, Providence, 2002.

[WBBP06]   J. Weickert, A. Bruhn, T. Brox, and N. Papenberg. A survey on variational methods for small displacements. In O. Scherzer, editor, *Mathematical Models for Registration and Applications to Medical Imaging*, volume 10 of *Mathematics in Industry*. Springer, Berlin, 2006.

[WBPB03]   J. Weickert, A. Bruhn, N. Papenberg, and T. Brox. Variational optic flow computation: from continuous models to algorithms. In L. Alvárez, editor, *Proc. International Workshop on Computer Vision and Image Analysis*, volume 26 of *Cuadernos del Instituto Universitario de Ciencias y Tecnologias Ciberneticas*, pages 1–6, Las Palmas, Spain, September 2003.

[Wei94a]   J. Weickert. Anisotropic diffusion filters for image processing based quality control. In A. Fasano and M. Primicerio, editors, *Proc. Seventh European Conference on Mathematics in Industry*, pages 355–362. Teubner, Stuttgart, 1994.

[Wei94b]   J. Weickert. Scale-space properties of nonlinear diffusion filtering with a diffusion tensor. Technical Report 110, Laboratory of Technomathematics, University of Kaiserslautern, Germany, October 1994.

[Wei98]   J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart, 1998.

[Wei99]   J. Weickert. Coherence-enhancing diffusion of colour images. *Image and Vision Computing*, 17(3–4):199–210, March 1999.

[Wes92]   P. Wesseling. *An Introduction to Multigrid Methods*. Wiley, Chichester, 1992.

[WHS+01]   J. Weickert, J. Heers, C. Schnörr, K. J. Zuiderveld, O. Scherzer, and H. S. Stiehl. Fast parallel algorithms for a broad class of nonlinear variational diffusion approaches. *Real-Time Imaging*, 7(1):31–45, February 2001.

[WPFS05] S. Wirtz, N. Papenberg, B. Fischer, and O. Schmidt. Robust and staining-invariant elastic registration of a series of images from histologis slices. In J. M. Fitzpatrick and J. M. Reinhardt, editors, *Medical Imaging: Image Processing*, volume 5747 of *Proceedings of SPIE*, pages 1256–1262, April 2005.

[WS99]   J. Weickert and C. Schnörr. Räumlich–zeitliche Berechnung des optischen Flusses mit nicht-linearen flußabhängigen Glattheitstermen. In W. Förstner, J.M. Buhmann, A. Faber, and P. Faber, editors, *Mustererkennung 1999*, pages 317–324. Springer, Berlin, 1999.

[WS01a]   J. Weickert and C. Schnörr. A theoretical framework for convex regularizers in PDE-based computation of image motion. *International Journal of Computer Vision*, 45(3):245–264, December 2001.

[WS01b]   J. Weickert and C. Schnörr. Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14(3):245–255, May 2001.

[WWB88]   A. M. Waxman, J. Wu, and F. Bergholm. Convected activation profiles and the measurement of visual motion. In *Proc. 1988 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 717–723. IEEE Computer Society Press, 1988.

[YD06]   I. Yavneh and G. Dardyk. A multilevel nonlinear method. *SIAM Journal on Scientific Computing*, 28(1), 2006. To appear.

[You71]   D. M. Young. *Iterative Solution of Large Linear Systems*. Academic Press, New York, 1971.

[ZSL97]   G. Zini, A. Sarti, and C. Lamberti. Application of continuum theory and multi-grid methods to motion evaluation from 3D echocardiography. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 44(2):297–308, March 1997.