Technische Universität München

Computational Science and Engineering (Int. Master's Program)

Master's Thesis

Combining Variational and Feature-based Methods for Motion Estimation

Levi Valgaerts

First examiner:

Prof. Joachim Weickert

Second examiner:

Prof. Nassir Navab

Assistant advisor:

Dr. Andrés Bruhn

Thesis handed in on: 31/01/2007

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

Saarbrücken,

Zusammenfassung

Variationsansätze zählen zur Zeit zu den genauesten Verfahren zur Berechnung des optischen Flusses aus Bildfolgen. In den letzten Jahen wurden dabei sowohl im Bezug auf die Genauigkeit als auch die Echtzeitfähigkeit grosse Fort-schritte gemacht. Insbesonder die Modellierung neuer Daten- und Glattheitsterme innerhalb der Energiefunktionale führte zur einer erheblichen Verbesserung der Schätzergebnisse. Es wurden jedoch bis jetzt kaum Anstrengungen unternommen, dichte variationelle optische Flussverfahren mit nicht-dichten merkmalsbasierten Methoden wie SIFT zu kombinieren. Das Ziel dieser Arbeit ist zu untersuchen, wie und in welchem Maße variationelle optische Flussverfahren von den Ergebnissen eines Merkmalsextraktors profitieren können. Dazu stellen wir eine neues Verfahren zur Bestimmung des optischen Flusses vor, das im wesentlichen auf der Interpolation vektorwertiger Daten beruht. Im Gegensatz zu üblichen Variationsansätzen beizieht dieses Verfahren ein durch das Matching einiger signifikatner Merkmale erhaltenes nicht-dichte Verschiebungsvektorfeld in die Berechnung des optischen Flusses mit ein. Dabei ermöglichen Ähnlichkeiten in der Formulierung des Variationsansatzes und des merkmalsbasierten Verfahrens die Verschmelzung der beiden Methoden zu einem hybriden Ansatz. Die Genauig-keit der beiden einzelen Verfahren sowie des kombinierten Ansatzes wird schließ-lich ein einer Reihe von Experimenten verglichen.

Abstract

Variational methods currently belong to the most accurate techniques for the recovery of the optic flow of an image sequence. Major contributions have been made in the last years to both their accuracy and their real-time performance. Most of the efforts to improve the accuracy of these methods have focused on the design of new data terms and smoothness terms that enter the energy functional. So far no research has been done on combining variational optic flow with feature-based methods such as SIFT. The goal of this thesis is to investigate how and to what extend optic flow methods can benefit from the results of a feature extractor. We will introduce a new method for optic flow estimation that is based on a unified model for tensor field interpolation. This method uses the sparse displacement field, that can be established by matching features from two images, to estimate the dense optic flow. Because of their similar formulation, the variational method and the feature-based method can easily be merged into one combined method. The accuracy of the three methods is finally compared in a series of experiments.

Acknowledgements

The work presented here could never have been accomplished without the contributions of many other people.

First of all I thank Dr. Andrés Bruhn for supervising this thesis and for spending much of his time on proof reading many draft versions.

I owe my gratitude to Prof. Joachim Weickert for giving me a chance to take part in his research.

I would like to thank all the members of the Mathematical Image Analysis Group at Saarland University for providing an excellent working atmosphere. In particular I would like to thank Michael Brueß, Stephan Didas, Luis Pizarro and Martin Welk for many insightful discussions.

I thank Prof. Nassir Navab and Dr. Selim Benhimane at the Technical University of Munich for their support and for introducing me to the field of computer vision.

I thank all my colleagues of the Computational Science and Engineering program in Munich for a wonderful two years, in particular Veselin Dikov, Niko Manopulo, Ilya Saverchenko, Gordana Stojceska and Ruben Sousa de Leon.

I thank Noreen Nasar for her company.

I thank my parents for supporting me and always believing in me.

Last but not least I would like to express my most profound gratitude to Jasmina Bogojeska for her unlimited support and patience, especially during the last weeks of the completion of this thesis.

Contents

Chapter 1

Introduction

The 2D displacement field that describes the motion of a scene between two consecutive frames of an image sequence is called *optic flow*. Determining the optic flow is an important correspondence problem in computer vision with many fields of application. Robot navigation and tracking rely on the ability to extract accurate motion information of objects within view of the camera. Stereo reconstruction and image registration are directly related to optic flow calculation in that they too try to establish a mapping between two or more images.

Variational Optic Flow Methods

Variational methods belong today to the most accurate and best understood methodologies for computing the optic flow. Such methods determine the unknown displacement field as the minimizer of an energy functional that penalizes deviations from model assumptions. Ever since the seminal work of Horn and Schunck (see [25]), this energy functional is typically made up of two terms: a data term that imposes constancy on specific image features, and a smoothness term that in general imposes additional constraints on the roughness of the solution. The smoothness term fills in information in regions where the data term does not have a unique solution (c.f. the aperture problem [4]) and provides a dense flow estimation. Because of this quality, variational methods are termed global methods in contrary to local differential methods that minimize local expressions as for instance the Lucas-Kanade method (see [29]).

There has been a major drive from within the community over the last years in the improvement of both preciseness and computational efficiency of variational optic flow estimation. New variational models with refined data constraints and new smoothness terms have led to the development of a toolkit for the design of highly accurate methods (see [35, 6, 31, 32, 14, 7, 36, 8]). The usage of fast numerical algorithms like multigrid allow for real-time employment (see [11, 10, 12]).

Hybrid Optic Flow Methods

On the other side of the correspondence spectrum we find feature-based approaches. These techniques extract a number of salient image locations in two images and try to match them. Thereby they establish a sparse set of correspondences. In particular in the field of stereo reconstruction such feature-based methods have been very popular (see [18]). Well known image features are edges and corners and many algorithms have been proposed in literature to obtain unique and unambiguous matches. Recently the Scale Invariant Feature Transform (SIFT) has been introduced (see [28]) and due to its superior performance it has become a predominant tool for fea-

ture extraction. The reason for this is that SIFT points are very stable and can be matched distinctively because of the association with a high dimensional descriptor. SIFT has been successfully utilized for object recognition and the construction of panoramas from multiple views.

So far not much attention has been devoted to the combination of local feature-based approaches and global variational methods for optic flow estimation. With the optic flow method of Brox *et al.* (see [7]), the pinnacle of accuracy of variational methods seemed to be reached. The question remained open if a hybrid method, that incorporates the sparse displacement field obtained by a feature-based technique, could improve upon the established results.

In this work we will present a global feature-based optic flow method that is founded on the model for vector field approximation from [44]. Although the method does not rely on the specific choice of feature extractor, we restrict ourselves in our evaluation to SIFT. The feature-based approximation method will be combined with a variational optic flow method to give rise to a hybrid model that takes advantage of the best of both worlds. This is close in spirit to the approaches in [13, 14] where local differential models have been embedded in variational optic flow techniques.

Outline

This thesis is organized as follows:

We begin in Chapter 2 by briefly summarizing the basics of variational optic flow estimation. The general structure of the energy functional and the necessary condition for a minimizer, i.e. the Euler-Lagrange equations, are discussed. We will present choices for data and smoothness terms and shortly make the connection between vector-valued diffusion processes and the induced filling-in effect.

In Chapter 3 we introduce our feature-based optic flow model. After an overview of types of interest point and the SIFT algorithm, we move on to the data approximation paradigm. Analogous to variational optic flow, different choices for the smoothness constraint are presented, each one with specific interpolation qualities. We conclude the chapter with ideas for dealing with outlying data in the sparse displacement set.

In Chapter 4 a combined optic flow method is presented. The system of partial differential equations (PDEs) that describes the feature-based model and the Euler-Lagrange equations of the variational model will be brought together to form a novel hybrid approach.

Chapter 5 approaches the task of discretizing the systems of PDEs that describe the three optic flow methods. We discuss the resulting large system of equations and propose techniques for efficiently solving it.

In Chapter 6 we deliver an experimental comparison of all three models presented in this work. Preliminary tests are conducted in which we use ground truth data instead of actual feature displacements to obtain an upper limit for potential improvement. Further on, SIFT is used for the generation of a sparse displacement field to illustrate that our feature-based approach works as a stand-alone method; comparisons with the combined method are carried out. In some final test runs we assess various outlier removal strategies.

We close with conclusions and future work in Chapter 7.

Naming and Notation

Under a *standard* optic flow estimation we will understand the solution of a system of PDEs with terms that arise from a constancy assumption and terms that arise from a smoothness constraint. Depending on the type of smoothness constraint a corresponding energy functional can or cannot be defined. Some of the standard optic flow methods that are included in our experiments do not minimize an associated energy functional and can therefore not be called variational methods.

Scalars will be written in italic as x. Vectors are column matrices and will be written in bold face as $\mathbf{x}$. Matrices and sets will be denoted by capital letters as A. The notations f_x and $\partial_x f$ will be used interchangeably for the partial derivative of f with respect to x.

Chapter 2

Variational Optic Flow Methods

Variational optic flow methods recover the displacement field of an image sequence as the minimizer of a suitable energy functional. Analogous to variational image regularization this functional consists of two terms: a *data term* and a *smoothness term*. The data term is responsible for the generation of information while the smoothness term allows for propagation of information. Variational optic flow methods are global methods that yield dense flow fields. This means that the smoothness term dominates the data term in the absence of sufficient local gradient information and that information is filled in from surrounding regions. In contrast to local methods, such as the Lucas-Kanade approach (see [29]), there is no need for a postprocessing step where sparse data have to be interpolated.

In this chapter we offer a short overview of the basic elements of variational optic flow methods. We will start by introducing the general structure of the energy functional and the necessary conditions that must be fulfilled by any minimizer. The next two sections will address design options for both the data term and the smoothness term. We introduce the structure tensor and we will point out how various smoothness terms can be classified based on the type of diffusion they induce. The list of smoothing operators presented here is not complete since we restrict ourselves to diffusion processes that arise from energy functionals. In the next chapter we will review the smoothness constraint in the setting of data approximation and we will add new operators based on their interpolation qualities. For a detailed survey on the various data terms and the taxonomy of smoothness terms for variational optic flow, the reader is directed to [42] and [43].

2.1 General Structure

In a continuous framework we denote by $f(x, y, t)$ a scalar-valued image sequence, where (x, y) is the location and t is the time. In most cases f will be obtained by convolving an initial image sequence f_0 with a Gaussian kernel

$$K_\sigma(\mathbf{x}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|\mathbf{x}|^2}{2\sigma^2}\right)$$

with standard deviation σ:

$$f = K_\sigma * f_0.$$

In this context σ can be regarded as a noise scale because the low-pass effect of Gaussian convolution attenuates high frequency noise, making the signal more suitable for operations like differentiation. The optic flow $\mathbf{w}(x,y,t) = (u(x,y,t), v(x,y,t), 1)^\top$ gives the displacement rate between subsequent frames with a temporal frame distance chosen to be 1. In a variational setting we are looking for the optic flow $\mathbf{w}$ that minimizes a continuous energy functional of the form

$$E(\mathbf{w}) = \int_\Omega \big(M(D^k f, \mathbf{w}) + \alpha \cdot S(\nabla f, \nabla \mathbf{w})\big)\, d\mathbf{x}. \tag{2.1}$$

The functional $E(\mathbf{w})$ penalizes deviations from model assumptions. It typically consists of a *data term* $M(D^k f, \mathbf{w})$, where $D^k f$ describes the set of all partial derivatives of f, and a *regularizer* or *smoothness term* $S(\nabla f, \nabla \mathbf{w})$. The data term is mostly expressed as a constancy assumption while the smoothness term penalizes deviations from (piecewise) smoothness. The parameter $\alpha > 0$ is a regularization parameter and serves as a weight between the data term and the smoothness term. The larger we choose the regularization parameter, the more simplified the resulting flow field will be. The optimal choice for α depends on the image sequence and on the combination of data and smoothness term because both terms commonly work on different scales of magnitude. The integration domain Ω can be either a spatial or a spatiotemporal domain. In the former case we set $\mathbf{x} = (x,y)^\top$ and $\nabla := \nabla_2 := (\partial_x, \partial_y)^\top$, while in the latter case $\mathbf{x} = (x,y,t)^\top$ and $\nabla := \nabla_3 := (\partial_x, \partial_y, \partial_t)^\top$. Imposing extra temporal smoothness will on the main give better results but we will restrict ourselves here to spatial models.

If the functional $E(\mathbf{w})$ is strictly convex, there is a unique minimizer $(u,v)^\top$. It can be shown (see [21]) that this minimizer is the solution of a system of partial differential equations (PDEs). These PDEs are called the *Euler-Lagrange equations* and are a necessary condition for each minimizer of $E(\mathbf{w})$. For our energy functional (2.1) the Euler-Lagrange equations take on the form

$$\partial_u M - \alpha \cdot \big(\partial_x S_{u_x} + \partial_y S_{u_y} \big) = 0, \tag{2.2}$$

$$\underbrace{\partial_v M}_{\text{data}} - \alpha \cdot \underbrace{\big(\partial_x S_{v_x} + \partial_y S_{v_y} \big)}_{\text{smoothness}} = 0, \tag{2.3}$$

with homogeneous Neumann (reflecting) boundary conditions

$$\begin{aligned} \partial_{\mathbf{n}} u = \mathbf{n}^\top u = 0 \\ \partial_{\mathbf{n}} v = \mathbf{n}^\top v = 0 \end{aligned} \qquad \text{on } \partial\Omega$$

where $\mathbf{n}$ is the outer normal vector on $\partial\Omega$.

2.2 Data Terms

Many differential methods for optic flow estimation are based on the assumption that the brightness f does not change along the path of motion $(x(t), y(t))^\top$ i.e. that the grey value of objects does not vary over time. Considering the instantaneous rate of change in brightness we can write this formally as

$$\frac{df\,(x(t), y(t), t)}{dt} = 0.$$

After applying the chain rule this yields the following optic flow constraint (see [29, 25]):

$$f_x u + f_y v + f_t = 0, \tag{2.4}$$

where $(u, v, 1)^\top = (\partial_t x, \partial_t y, 1)^\top$ is the desired rate of displacement. This expression can also be regarded as a first order Taylor approximation in the point (x, y, t) of the brightness constancy constraint between two subsequent frames:

$$f(x + u, y + v, t + 1) - f(x, y, t) = 0.$$

The Taylor linearization is only a reasonable approximation if the displacements are small and if the flow field is sufficiently smooth. Equation (2.4) is often used in the functional (2.1) with a quadratic penalization:

$$M(D^1 f, \mathbf{w}) = (\mathbf{w}^\top \nabla_3 f)^2. \tag{2.5}$$

This squared constraint can conveniently be written down as follows:

$$\begin{aligned} M &= (\mathbf{w}^\top \nabla_3 f)^2 \\ &= \mathbf{w}^\top (\nabla_3 f \, \nabla_3 f^\top) \, \mathbf{w} \\ &= \mathbf{w}^\top J_0(\nabla_3 f) \, \mathbf{w}, \end{aligned} \tag{2.6}$$

where $J_0(\nabla_3 f) = \nabla_3 f \, \nabla_3 f^\top$ is a 3×3 symmetric positive semidefinite matrix. The brightness constancy assumption enters the earliest representative of the class of variational optic flow methods, *the method of Horn and Schunck* (see [25]). This energy functional reads:

$$E_{HS}(\mathbf{w}) = \int_\Omega \left(\mathbf{w}^\top J_0(\, \nabla_3 f \,) \, \mathbf{w} + \alpha \cdot |\nabla \mathbf{w}|^2 \right) d\mathbf{x}. \tag{2.7}$$

As a smoothness term the homogeneous regularizer $|\nabla \mathbf{w}|^2 = |\nabla u|^2 + |\nabla v|^2$ is used which can be directly related to homogeneous diffusion filtering (see [43]). The Euler-Lagrange equations corresponding to $E_{HS}(\mathbf{w})$ are given by the system of PDEs:

$$f_x^2 u + f_x f_y v + f_x f_t - \alpha \cdot \Delta u = 0, \tag{2.8}$$

$$f_x f_y u + f_y^2 v + f_y f_t - \alpha \cdot \Delta v = 0, \tag{2.9}$$

with reflecting boundary conditions. When we denote by J_{mn} the entry of $J_0(\, \nabla_3 f \,)$ on the mth row and the nth column, we can write equations (2.8)-(2.9) as:

$$J_{11} u + J_{12} v + J_{13} - \alpha \cdot \Delta u = 0, \tag{2.10}$$

$$J_{12} u + J_{22} v + J_{23} - \alpha \cdot \Delta v = 0, \tag{2.11}$$

where $J_{12} = J_{21}$ because J_0 is symmetric.

The Horn and Schunck method can be extended in a natural way by replacing the tensor J_0 by the spatiotemporal *structure tensor* (see [19])

$$J_\rho(\nabla_3 f) = K_\rho * J_0(\nabla_3 f), \tag{2.12}$$

where $\rho > 0$ is an integration scale which provides an averaging of the gradient directions in J_0. This extension is coined the combined local-global (CLG) method (see [14]) since it combines the noise robustness of local methods with the dense estimation of global approaches. The CLG method minimizes the functional

$$E_{CLG}(\mathbf{w}) = \int_\Omega \left(\mathbf{w}^\top J_\rho(\, \nabla_3 f \,) \, \mathbf{w} + \alpha \cdot |\nabla \mathbf{w}|^2 \right) d\mathbf{x}. \tag{2.13}$$

Besides $J_\rho(\nabla_3 f)$, also the spatial structure tensor $J_\rho(\nabla_2 f) = K_\rho * (\nabla_2 f \, \nabla_2 f^\top)$ (see [19]) plays an important role in image processing and is often utilized to analyze local image structure. $J_\rho(\nabla_2 f)$ is a symmetric positive semidefinite matrix of which the eigenvectors indicate the preferred local structure directions and the eigenvalues the contrast along these directions. We will see in Chapter 3 that the spatial structure tensor can be employed to determine whether a certain image location falls in a smooth area or if it is part of an edge or a corner.

Throughout this thesis we will mainly make use of the brightness constancy assumption because it is the most simple one. However other data terms are available. In practice the choice of which data term to use in optic flow estimation depends on prior knowledge about the scene and on assumptions about which image features are invariant under the considered motion. Data term (2.5) gives good results as long as there are no illumination changes present in the image sequence. If this assumption is violated we have to look for other image features on which we can impose constancy. One such feature is the spatial brightness gradient $(f_x, f_y)^\top$. Assuming that the gradient does not change along the path of motion we end up with the data term (see [7])

$$M(D^2 f, \mathbf{w}) = (\mathbf{w}^\top \nabla_3 f_x)^2 + (\mathbf{w}^\top \nabla_3 f_y)^2. \tag{2.14}$$

We can further impose constancy on higher-order derivatives like the entries of the spatial Hessian $\mathcal{H}_2 f$ (see [36]). This results in an expression involving all second order spatial derivatives. We can also devise terms that do not contain any directional information and are therefore fit for any type of motion, including fast rotation. Such data terms would impose constancy on features like the gradient magnitude $|\nabla_2 f|$, the Laplacian $\Delta_2 f$ or the determinant of $\mathcal{H}_2 f$ (see [42]). Additionally results can be improved by linearly combining constancy constraints on several image features $p_1, ..., p_n$. As an extension of (2.6), the data term can then be written as a quadratic form

$$M = \mathbf{w}^\top J(\nabla_3 p_1, ..., \nabla_3 p_n)\, \mathbf{w},$$

where

$$J(\nabla_3 p_1, ..., \nabla_3 p_n) = \sum_{i=1}^{n} \gamma_i \nabla_3 p_i \nabla_3 p_i^\top.$$

The matrix $J(\nabla_3 p_1, ..., \nabla_3 p_n)$ is called the motion tensor that is derived from the weighted sum of n squared linearized constraints with positive weights γ_i (see [9]). As with the structure tensor we can perform a weighted least squares fitting by convolving J with a Gaussian K_ρ. This makes the data term more robust under noise and outliers, such as occlusion areas.

Another way of increasing the robustness of the data term is to penalize deviations from constancy assumptions in a non-quadratic way, thereby giving outliers less weight (see [5]). Such a data term would take on the form

$$M = \Phi(\mathbf{w}^\top J_\rho \, \mathbf{w}), \tag{2.15}$$

where $\Phi(s^2)$ is a convex function in s. The penalizer $\Phi(s^2)$ is chosen to be convex such that the resulting energy functional has a unique minimum and allows for the construction of globally convergent algorithms. An example is the regularized L_1-penalization which will be introduced in section 3.5.

2.3 Smoothness Terms

The smoothness term is responsible for the propagation of information to locations where no unique solution for the data term exists. Different models for the regularizer are available depending on the desired type of filling-in. To make their effects comprehensive, it has been shown in [43] that it is helpful to relate them to diffusion processes of vector-valued images. Following this idea we regard the Euler-Lagrange equations (2.2)-(2.3) as the steady-state conditions of a diffusion-reaction system in which the diffusion term results from $S(\nabla f, \nabla \mathbf{w})$ and the reaction term from $M(D^k f, \mathbf{w})$. The strong connection between regularization and diffusion filtering becomes clear when we restrict ourselves to the diffusion part of this diffusion-reaction system:

$$\partial_t u = \partial_x S_{u_x} + \partial_y S_{u_y}, \tag{2.16}$$

$$\partial_t v = \partial_x S_{v_x} + \partial_y S_{v_y}. \tag{2.17}$$

We can interpret these equations as a diffusion filter of some vector-valued image with channels u and v. In the classification process we identify the optic flow regularizers $S(\nabla f, \nabla \mathbf{w})$ that produce the different types of multichannel diffusion. It has to be noted that this taxonomy, introduced in [9], is based on diffusion processes that can be written in divergence form and can be regarded as the minimizer of a suitable energy functional.

- Homogeneous Regularization. Homogeneous diffusion of vector-valued images corresponds to *homogeneous* regularization in optic flow. As we have seen with the Horn and Schunck method, the according regularizer can be written as $S(\nabla \mathbf{w}) = |\nabla \mathbf{w}|^2 = |\nabla u|^2 + |\nabla v|^2$ and the Euler-Lagrange equations as (2.10)-(2.11). Motion boundaries will in general get blurred and dislocated when we use the homogeneous smoothness term.

- Flow-driven Isotropic Regularization. Homogeneous regularization does not preserve edges in the flow field and as a consequence sudden motion transitions are estimated inaccurately. To overcome this unwanted effect, discontinuity preserving regularizers have been developed that inhibit smoothing at edge locations. In the case of *flow-driven isotropic* regularization the resulting diffusion process is called nonlinear isotropic diffusion and it uses a scalar-valued diffusivity $g(s^2)$ that is decreasing in s. The corresponding optic flow regularizer is $S(\nabla \mathbf{w}) = \Psi(|\nabla u|^2 + |\nabla v|^2)$ where $\Psi' = g$. For a certain data term M this leads to the Euler-Lagrange equations:

$$\partial_u M - \alpha \cdot \operatorname{div}\left(g(|\nabla u|^2 + |\nabla v|^2|)\, \nabla u\right) = 0, \tag{2.18}$$

$$\partial_v M - \alpha \cdot \operatorname{div}\left(g(|\nabla u|^2 + |\nabla v|^2|)\, \nabla v\right) = 0. \tag{2.19}$$

This type of regularization is called flow-driven because there is a feedback of the evolving flow field $\mathbf{w}$ into the process through the diffusivity g. If the gradient $\nabla \mathbf{w}$ is large, the diffusivity g will be close to 0, thus not allowing any smoothing.

- Flow-driven Anisotropic Regularization. Using a diffusion tensor D instead of a scalar-valued diffusivity amounts to direction dependent behavior. D can be designed such as to allow diffusion along flow edges and inhibit diffusion across them. This can improve the results of flow estimation at motion boundaries, especially in the presence of noise. *Flow-driven anisotropic* regularization results in nonlinear anisotropic diffusion that makes use of a diffusion tensor that is a function of $\mathbf{w}$.

A straightforward choice for D would be to specify its eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ as those of the joint structure tensor $\nabla u \nabla u^\top + \nabla v \nabla v^\top$ and its eigenvalues as $g(\mu_1)$ and $g(\mu_2)$. Here μ_1 and μ_2 are the eigenvectors of the joint structure tensor and g a scalar-valued diffusivity. For this type of diffusion a formal expression for the regularizer only exists when the flow components u and v are not presmoothed. The Euler-Lagrange equations are:

$$\begin{aligned} \partial_u M - \alpha \cdot \operatorname{div}\left(D(\nabla u \nabla u^\top + \nabla v \nabla v^\top)\nabla u\right) &= 0, & (2.20) \\ \partial_v M - \alpha \cdot \operatorname{div}\left(D(\nabla u \nabla u^\top + \nabla v \nabla v^\top)\nabla v\right) &= 0. & (2.21) \end{aligned}$$

- Image-driven Isotropic Regularization. If motion boundaries are a subset of the image boundaries, we can prevent smoothing at motion boundaries by introducing a weight function that becomes small at image edges. Just as with flow-driven isotropic regularization we use a scalar-valued diffusivity $g(s^2)$, but this time g is in function of the image gradient $|\nabla f|$. This yields the regularizer $S(\nabla f, \nabla \mathbf{w}) = g(|\nabla f|^2)(|\nabla u|^2 + |\nabla v|^2)$ with Euler-Lagrange equations:

$$\begin{aligned} \partial_u M - \alpha \cdot \operatorname{div}\left(g(|\nabla f|^2)\,\nabla u\right) &= 0, & (2.22) \\ \partial_v M - \alpha \cdot \operatorname{div}\left(g(|\nabla f|^2)\,\nabla v\right) &= 0. & (2.23) \end{aligned}$$

- Image-driven Anisotropic Regularization. If we make the diffusion tensor D for optic flow regularization dependent on the image gradient $|\nabla f|$, we end up with *image-driven anisotropic regularization*. It leads to the Euler-Lagrange equations:

$$\begin{aligned} \partial_u M - \alpha \cdot \operatorname{div}\left(D(\nabla f)\nabla u\right) &= 0, & (2.24) \\ \partial_v M - \alpha \cdot \operatorname{div}\left(D(\nabla f)\nabla v\right) &= 0. & (2.25) \end{aligned}$$

By basing the design of the type of smoothing solely on the diffusion equations (2.16)-(2.17) we can choose from other models than the ones presented here, depending on the resulting kind of filling-in effect. One has to take into account that not for all models a corresponding energy functional can be defined such that the derived Euler-Lagrange equations give a system of steady-state diffusion-reaction equations.

Chapter 3

Feature-based Optic Flow Methods

Determining a correspondence between two or more images can be done by extracting a number of salient local image features and subsequently matching them to each other. The discrete correspondences found in this way are used in many computer vision tasks, ranging from object recognition to image registration and 3D reconstruction. Under feature or interest point extraction one most often understands the selection of distinctive locations in the image where there is a strong change in the grey value. Classical examples of locations of interest are edges and corners. More recently new scale invariant features with high repeatability were introduced to the field. Among these the so called SIFT features have been shown to outperform other local descriptors and are currently widely used.

In this chapter we demonstrate how a set of feature matches can be employed to estimate the dense motion field from an image sequence. The first two sections give a short description of feature extraction and the SIFT algorithm. Section 3.3 defines the problem of dense motion estimation from discrete point matches and the subsequent sections treat possible solutions. In the next chapter we will show ways of incorporating the information from point matches into standard variational optic flow methods.

3.1 Feature Extraction and Matching

Edges and corners are common feature types and extracting them from image data is a key task in computer vision. Edges are detected with derivative operators. Since taking derivatives is an ill-posed problem, the original grey value image f is convolved with a Gaussian kernel K_σ to obtain a smoother image $u = K_\sigma * f$. In its simplest form the gradient magnitude $|\nabla u|$ is approximated with Sobel operators and edges are located where $|\nabla u|$ exceeds a certain threshold. A more sophisticated version of this approach that thins down the edges to a width of 1 pixel was presented by Canny in [15]. Edges can also be defined as zero crossings of the Laplacian of Gaussians (LoG) Δu (see [30]). This detector yields closed contours, but can give incorrect edge locations in case of strong Gaussian smoothing.

Corners can be robustly found by using the information about the local image structure encompassed in the structure tensor $J_\rho(\nabla_2 u)$ (see Chapter 2). The eigenvectors of J_ρ specify the preferred local structure directions, while the eigenvalues λ_1 and λ_2 describe the average contrast along these directions. Very generally, corners can be described as locations where $\lambda_1 \geq \lambda_2 \gg 0$. A popular variant is the

Harris corner detector (see [22]) which evaluates the expression

$$\det J_\rho - k \operatorname{tr}^2 J_\rho \stackrel{!}{=} \max \tag{3.1}$$

for a certain value of k, where max is the maximum in a local neighborhood. A similar detector by Rohr only looks for local maxima of $\det J_\rho$. Other approaches make use of second order derivatives or embed the corner detection into a morphological scale space to avoid the dislocating effects of Gaussian smoothing.

Once features have been extracted in both images, correspondences have to be established between them. Formally we have to find a set of n feature matches $\{\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i \,|\, i = 1, \ldots, n\}$ given a set of features $\{\mathbf{x}_i \,|\, i = 1, \ldots, p\}$ and $\{\mathbf{x}'_i \,|\, i = 1, \ldots, q\}$ in the first and in the second image, respectively. In [2] the authors include an optic flow technique which is based on tracking an edge map throughout an image sequence. In practice though, it is more common to look for correspondences between point features like corners. In this case an initial set of point matches can be found by correlating the image patches surrounding a point $\mathbf{x}_i$ and a point $\mathbf{x}'_i$. A threshold is then applied to the correlation score to obtain a number of possible correspondences. These correspondences can be ambiguous since one point might be paired to several other points. In [47] a relaxation scheme relying on a measure for the strength of a match is proposed in order to overcome these ambiguities. Matching image patches by simple correlation is very sensitive to viewpoint changes or deformations. Therefore, recent matching techniques make use of a so called *descriptor*, an adequate vector representation of the neighborhood of a point of interest. Matching feature points is based on a comparison of a distance measure between two descriptor vectors, e.g. the Euclidian or the Mahalanobis distance. Among these last approaches the *Scale Invariant Feature Transform* has proven to be the most successful.

3.2 Scale Invariant Feature Transform

Repeatability is an important property of a feature detector. This means that under different viewing conditions preferably the same interest points are to be extracted. The set of features found for instance by the Harris corner detector depends strongly on the noise scale σ and the integration scale ρ that are used in the calculation of J_ρ. A simple Harris corner detector is therefore not invariant to image scale. In [28] Lowe introduced a detector-descriptor-method that is not only scale invariant but also invariant with respect to image rotation and to a certain extent with respect to affine distortions and illumination changes. The method, the Scale Invariant Feature Transform (SIFT), generates features that are highly distinctive in the sense that they can be correctly matched to one other with high probability. Recent comparative studies by Mikolajczyk *et al.* (see [33, 34]) put forward SIFT as the most accurate matching algorithm.

A variant of SIFT, PCA-SIFT (see [26]), speeds up matching by reducing the dimension of the descriptor space. We experienced , however, that the performance of PCA-SIFT was not better than the performance of SIFT and this is also confirmed in one of the above mentioned studies. Of another recently proposed detector-descriptor-method, SURF (see [3]), no performance evaluations are known yet, so we did not include it in our experiments.

3.2.1 The SIFT algorithm

The SIFT algorithm proceeds in two main stages:

1. **Detection:** Interest points are identified over all image scales and locations. Every point is assigned one or more orientations based on the prevalent local gradient directions.

2. **Description:** The distribution of image gradients within a neighborhood of each feature is collected into a representation. This operation is performed on image data that has been transformed relative to the assigned orientation, scale and location for each feature, thus providing invariance to these transformations.

Both stages will be explained in more detail below.

In a first step extrema are detected in Gaussian scale-space. The scale-space concept (see [45, 27]) can be defined as the embedding of an image $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$ into an ordered family

$$\{T_\sigma f | t \geq 0\}$$

of gradually smoothed, simplified versions, which satisfies certain requirements. For a Gaussian scale-space

$$T_\sigma f = K_\sigma * f$$

with K_σ a Gaussian with standard deviation σ. As stable features, maxima and minima of the normalized Laplacian of Gaussians (LoG) are chosen. Computational speed-up is acquired by approximating the LoG by the difference of Gaussians (DoG)

$$d(x, y, \sigma) = (K_{k\sigma} * f) - (K_\sigma * f)$$

where k is a factor separating two nearby scales. The DoG can be calculated by simply subtracting subsequent images that are sampled in scale-space. For this purpose a Gaussian pyramid of repeatedly convolved images is built where every image is downsampled in space after each octave. Selection of DoG extrema is done by comparing each sample point to all of its neighbors in space and scale and retaining those that are larger or smaller than all of them.

Subsequently the DoG is approximated in the retained sample points by its quadratic Taylor expansion. Fitting this function to the local neighborhood improves matching and stability and results in sub-pixel accuracy for the spatial feature coordinates. The minimizer of this Taylor approximation is taken as the location for the feature point. Points with low contrast and points that are poorly localized along edges are discarded.

Within a neighborhood of the found feature points the gradient orientations are sampled and added to a histogram. Each histogram entry is weighted by a Gaussian window and its gradient magnitude. Peaks in the histogram correspond to dominant directions of the local gradient and the according orientation is assigned to the feature. It is important to note that multiple peaks in the histogram lead to the creation of multiple feature points that can be matched against in the matching phase.

The last stage consists of building a descriptor vector for the local image region around each feature point. The descriptor is based on the distribution of gradient orientations since this should result in better matching under changes in 3D

viewpoint. The local image region is described with respect to the coordinate system imposed by the assigned location, scale and orientation, resulting in invariance to these parameters. From this region gradient magnitudes and orientations are sampled and weighted after which they are distributed over an array of orientation histograms. Finally, the descriptor vector is derived from the entries of the histogram bins and normalized to make it invariant to illumination changes. In the experiments conducted by Lowe the descriptor vectors had a length of 128, corresponding to a 4x4 array of histograms with each 8 bins. This high dimensionality is supported by the claim that the use of a lower dimensional descriptor space results in loss of accuracy.

3.2.2 Matching SIFT features

A SIFT point $\mathbf{x}_i$ is matched by identifying its nearest neighbor in descriptor space i.e. the SIFT point $\mathbf{x}'_i$ with the minimum Euclidean distance for the descriptor vector. Not every $\mathbf{x}_i$ in the first image has a corresponding match $\mathbf{x}'_i$ in the second image and vice versa. In the case of motion estimation this can be due to occlusions or because a feature in one image is simply not detected in the other image. In order to discard features that do not have any match the distance to the closest neighbor is compared to the one of the second closest neighbor. The rationale behind this is the following. To obtain reliable matching, a correct match will have to have its closest neighbor significantly closer than the closest incorrect match. A feature with no correct match will on the contrary have several neighbors that are similarly close in the high dimensional descriptor space. Lowe supports this conjecture with results from matching experiments involving a large database of features and a number of images that underwent a variety of transformations. He uses a distance ratio between the closest and the second closest point of not more than 0.8. An alternative is to use a threshold for the absolute distance to the closest point. In [26] such thresholds are determined from a database of various images. We have observed that this last approach does not result in a better quality of matches then the former approach.

The amount of feature matches obtained with the SIFT algorithm is relatively high and depends on the image and on the chosen threshold for the distance ratio. Highly textured regions give more features and thus more correspondences than regions with less gradient information. In most applications it is advantageous to have a large number of matches and this is also true for dense motion estimation. We obtained up to 600 point correspondences for a 316×252 image and more than 4100 for a 512×512 image.

3.3 Problem Statement

Let $f(x, y, t)$ be a scalar-valued image sequence where (x, y) is the location within the image domain $\Omega \in \mathbb{R}^2$ and t is the time. Let $f_1(x, y) = f(x, y, t)$ be a frame at time t and $f_2(x, y) = f(x, y, t + 1)$ be the next frame in the sequence. $\mathbf{x}_i$ and $\mathbf{x}'_i$ denote the location of a point of interest in f_1 and f_2. Having found a set of n feature matches $\{\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i \,|\, i = 1, \ldots, n\}$ between f_1 and f_2 we end up with a sparse displacement field

$$F = \{\mathbf{d}_i = \mathbf{x}'_i - \mathbf{x}_i \,|\, i = 1, \ldots, n\},$$

consisting of n displacement vectors which are located in the points $\mathbf{x}_i$. We can regard F as a set of observed values of a more generic variable $\mathbf{d}(x, y)$.

The task consists of finding a dense flow field $\mathbf{w}(x, y)$ between f_1 and f_2 using the scattered data $\mathbf{d}_i$ in $\mathbf{x}_i$ for $i = 1, ..., n$. By applying our feature matching algorithm on every two consecutive frames of the image sequence $f(x, y, t)$ we want to be able to estimate the optic flow

$$\mathbf{w}(x, y, t) = \begin{pmatrix} u(x, y, t) \\ v(x, y, t) \\ 1 \end{pmatrix}.$$

3.4 Interpolation and Approximation

In this section we assume that the errors in F are mainly due to a bad localization of the SIFT features and that they are Gaussian distributed and small (≤ 3 pixels). Displacement data resulting from false matches, so called outliers, do not follow this error distribution and how to deal with them is the topic of section 3.5.

In computer vision applications errors in the measurement of position are mostly dealt with by finding least squares solutions or minimizing cost functions that at the same time estimate model parameters and correct the measurements. These methods often relate to a maximum likelihood estimate. In our case we are dealing with general motion estimation from scattered data. We do not make any strong model assumptions like pure translational or rotational motion, motion between two stereo views of the same scene or the assumption that all objects in the scene move in the same way. Therefore we do not have model parameters that are the arguments of least squares expressions or of cost functions to be minimized. The only assumption we want to make about the displacement field $\mathbf{w}$ is that it is *smooth* or at least *piecewise smooth*. Thus the problem of finding $\mathbf{w}$ from F comes down to scattered data interpolation with the only constraint that $\mathbf{w}$ is piecewise smooth. Most likely we do not want to fit our data exactly since then we will mainly capture the idiosyncracies of the sampled displacement set F. In order to end up with a reasonable flow estimation in all points of the domain we will have to find a *general trend* in the data. This means finding a suitable regression curve or function approximation.

Although many methods exist for function approximation, both parametric and non parametric, we choose to keep the solution in a variational setting. Like this we will be able to assess the smoothing properties of different regularizers and compare the results with those of other variational approaches.

3.4.1 The Approximation Problem

At first we only consider the scalar case. We are given n points $x_1 < x_2 < ... < x_n$ with known function values $f(x_1), ..., f(x_n)$ and we want to find a smooth function $u(x)$ that approximates the unknown $f(x)$ on the given domain. A very general class of solution methods are the so called regularization methods in which $u(x)$ is the solution of

$$\min_u \left[\sum_{i=1}^{n} \mathcal{L}\left(f(x_i), u(x_i)\right) + \alpha \cdot \mathcal{J}(u) \right] \tag{3.2}$$

where $\mathcal{L}(f(x_i), u(x_i))$ is a *loss function*, $\mathcal{J}(u)$ a *penalty functional* and α a *smoothness parameter*. The loss function is a measure for the closeness of the fit to the data, while the second term tries to stabilize the solution by penalizing for instance the roughness of the fit. Important approximation methods that fall into this framework are for example smoothing splines and thin-plate splines for higher dimensions. Under certain restrictions the solution $u(x)$ can also be written as an

expansion in radial basis functions $K(x, x_i)$. The latter method combines the local fitting properties of kernel methods with the flexibility of basis expansion methods. Commonly used are Gaussian radial basis functions $K_\sigma(x, x_i) = e^{-|x-x_i|^2/2\sigma^2}$ with window size σ centered in the points x_i. A detailed coverage of the above mentioned methods in a statistical learning framework is given in [24].

A widely used loss function is the residual sum of squares. With a penalization of the curvature for smoothness we obtain:

$$\min_u \left[\sum_{i=1}^{n} (u(x_i) - f(x_i))^2 + \alpha \int (\partial_{xx} u(x))^2 dx \right]. \tag{3.3}$$

If the smoothness parameter $\alpha = 0$, the solution $u(x)$ can be any function that interpolates the data. If $\alpha = \infty$, a least squares fit is obtained since in this case no non-zero second order derivative is tolerated. It can be shown (see [24]) that the solution to the particular problem (3.3) is a natural cubic spline with knots in $x_i,\ i = 1, ..., n$. We can then write the solution as

$$u(x) = \sum_{i=1}^{n} \theta_i B_i(x)$$

where $B_i(x)$ are cubic B-spline basis functions. The problem is reduced to solving (3.3) for the coefficients θ_i.

3.4.2 A PDE-based Model for Function Approximation

We would like to use a much more general approach than the one discussed before. We do not want to make use of an explicit expression for the function approximation and we do not want to restrict ourselves to a specific penalizer or to the squared error loss. We built our approximation schemes on the unified model presented by Weickert *et al.* in [44]. It is based on elliptic PDEs and it covers some of the methods presented before. In our exposition of the model we set out from spline interpolation as the minimization of a suitable energy functional.

We assume that $x_0 = 0$ and $x_n = 1$. We start by considering the spline interpolation problem as looking for a smooth function $u(x) : [0, 1] \to \mathbb{R}$ that is the minimizer of

$$E(u) = \int_0^1 (\partial_x^m u)^2 dx \tag{3.4}$$

with the constraints

$$u(x_i) = f(x_i) \quad i = 1, ..., n. \tag{3.5}$$

Note that with this model only spline interpolants of odd order can be obtained.

The Euler-Lagrange equation for (3.4),

$$(-1)^{m+1} \partial_{xx}^m u = 0$$

can be combined with the constraints (3.5) to a single linear PDE of order $2m$:

$$c(x) \cdot (u(x) - f(x)) - (1 - c(x)) \cdot (-1)^{m+1} \partial_{xx}^m u = 0$$

with

$$c(x) := \begin{cases} 1 & \text{if } x \in \{x_0, ..., x_n\} \\ 0 & \text{else.} \end{cases}$$

For $m > 2$ the maximum-minimum principle will not be satisfied in general. This means that the interpolating spline may not remain in the convex hull of the data resulting in over- and undershoots. For $m = 1$ we get linear interpolation with C^0 smoothness at the interpolation points. For $m = 2$ we have cubic spline interpolation with C^2-smoothness in the interpolation points.

Additionally we can take a look at the variational restoration of a noisy signal $f : [0, 1] \to \mathbb{R}$. We want to find a signal $u(x)$ that is the minimizer of a convex energy functional that rewards smoothness and similarity to $f(x)$:

$$E(u) = \int_0^1 \left(c \cdot (u - f)^2 + (1 - c) \cdot \Psi(u_x^2)\right) dx \tag{3.6}$$

with a constant weight $0 < c < 1$ and an increasing penalizing function $\Psi : [0, \infty) \to \mathbb{R}$. Examples of penalizer functions are the Tikhonov, the Charbonier or the total variation (TV) penalizer (see [16, 39]).

The minimizer $u(x)$ of (3.6) has to satisfy the Euler-Lagrange equation

$$c \cdot (u - f) - (1 - c) \cdot \partial_x \left(\Psi'(u_x^2)\, u_x\right) = 0, \tag{3.7}$$

with homogeneous Neumann boundary conditions.

From the previous considerations on spline interpolation and variational restoration, we can derive a unified model for interpolation and function approximation. Assume we are given some incomplete or noisy scalar data f. We then obtain an interpolation or processed signal $u(x)$ by solving

$$c(x) \cdot \underbrace{(u - f)}_{\text{similarity}} - (1 - c(x)) \cdot \underbrace{Lu}_{\text{smoothness}} = 0, \tag{3.8}$$

with homogeneous Neumann boundary conditions

$$\partial_{\mathbf{n}} u = \mathbf{n}^\top \nabla u = 0 \qquad \text{on } \partial\Omega,$$

where $c(x) : [0, 1] \to [0, 1]$ is a confidence function, L some elliptic differential operator and $\mathbf{n}$ the outer normal vector on $\partial\Omega$.

At locations where $c(x) = 0$ missing data is allowed to be filled in by the smoothness assumption. Where $0 < c(x) < 1$ we obtain an approximation of the original signal and at locations where $c(x) = 1$, $u(x) = f(x)$. In order to have a method for approximation of scattered data with simultaneous filling-in we choose

$$c(x) = \begin{cases} \in (0, 1) & \text{if } x \in \{x_0, ..., x_n\} \\ 0 & \text{else.} \end{cases} \tag{3.9}$$

In analogy with (3.7) the parameter c in the interpolation points can be interpreted as a measure for how much we want to smooth or *denoise* our data. Hence this model agrees with our need for removing possible high frequency noise in the displacement data due to localization errors of the feature detector.

The attractive aspect of this model is that it allows us to choose many possible operators L. If we choose $Lu = \partial_{xx} u$ we get the linear interpolant discussed before. In order to get smoother results higher order operators can be used. A violation of the maximum-minimum principle is however possible in this case, but as we will see further, we can avoid this problem by choosing second-order nonlinear anisotropic

operators instead. Additionally we acquire a rotationally invariant approximation technique.

We can rewrite equation (3.8) with the constraints (3.9) as follows:

$$(u - f) - \frac{1 - c(x)}{c(x)} \cdot Lu = 0,$$

and by setting

$$\alpha(x) = \frac{1 - c(x)}{c(x)}$$

we get

$$(u - f) - \alpha(x) \cdot Lu = 0. \tag{3.10}$$

To avoid difficulties in defining $\alpha(x)$ on the domain we write equation (3.10) as:

$$1_{\{x_0,...,x_n\}} \cdot (u - f) - \alpha \cdot Lu = 0 \tag{3.11}$$

with

$$1_{\{x_0,...,x_n\}}(x) = \begin{cases} 1 & \text{if } x \in \{x_0, ..., x_n\} \\ 0 & \text{if } x \notin \{x_0, ..., x_n\} \end{cases}$$

the indicator function of the set of interpolation points $\{x_0, ..., x_n\}$ and $\alpha \in (0, \infty)$. This notation makes it easier to interpret the problem in terms of one location independent smoothness weight α. Equation (3.11) also clearly resembles the Euler-Lagrange equations arising from the variational optic flow problem (see Chapter 2), but with the optic flow constraint replaced by the similarity constraint.

Finally, we wish to emphasize that equation (3.8) has not been derived directly as the Euler-Lagrange equation of a corresponding functional, since the factor $c(x)$ would cause difficulties in trying to find such a functional. We nevertheless point out the similarity between the derived formula (3.11) and the minimization problem (3.3). Although it has not been proven, we anticipate that a minimizer $u(x)$ of (3.3) is also a solution of (3.11) in the illustrative case of $Lu = \partial^2_{xx} u$.

3.4.3 Choices for the Differential Operator

From now on we consider the 2D case of finding a vector data approximation to the unknown optic flow field $\mathbf{w}(u(x, y, t), v(x, y, t))$ from our data set F of feature displacements. The set of feature displacement locations $\{\mathbf{x}_0, ..., \mathbf{x}_n\}$ will be denoted as X. For selecting an appropriate differential operator L we can look at the types of regularizer used in variational optic flow methods as well as at vector valued diffusion filtering (see [43, 42]). In what follows we sum up the smoothing operators that were used in our experiments.

- Homogeneous Operator. In analogy with *homogeneous* diffusion filtering and homogeneous regularization in optic flow estimation we can write formula (3.11) as:

$$1_X \cdot (u - d_u) - \alpha \cdot \Delta u = 0, \tag{3.12}$$

$$1_X \cdot (v - d_v) - \alpha \cdot \Delta v = 0, \tag{3.13}$$

where we denote by d_u and d_v the u- and v-component of $\mathbf{d}$. This is a generalization of piecewise linear fitting to the fitting of scattered 2D vector data and the equivalent of the method of Horn and Schunk. The result has to be expressed in terms of radial basis functions centered in the interpolation points. If we choose α close to 0 we almost obtain interpolation. If we choose α very large the solution will evolve towards a constant, the mean of F.

- Biharmonic Operator. Homogeneous diffusion as an interpolant tends to give singularities in the interpolation points. The use of higher order operators might avoid this problem. As a higher order operator we choose the *biharmonic* smoother:

$$1_X \cdot (u - d_u) + \alpha \cdot \Delta^2 u = 0, \tag{3.14}$$
$$1_X \cdot (v - d_v) + \alpha \cdot \Delta^2 v = 0. \tag{3.15}$$

This is a generalization of the 1D cubic spline fitting and is known as a thin plate spline with radial basis functions of the form $K(\mathbf{x}, \mathbf{x}_i) = |\mathbf{x} - \mathbf{x}_i|^2 \log(|\mathbf{x} - \mathbf{x}_i|)$. For increasing α the result will evolve towards a linear fit.

- Flow-driven Isotropic Operator. Both the above mentioned operators have the undesirable property that they smooth across motion discontinuities. Therefore motion boundaries can become blurry and less distinctive. *Nonlinear isotropic* diffusion avoids the blurring of motion edges by incorporating a diffusivity g under the divergence expression:

$$1_X \cdot (u - d_u) + \alpha \cdot \operatorname{div}\left(g(|\nabla u_\sigma|^2 + |\nabla v_\sigma|^2)\, \nabla u\right) = 0, \tag{3.16}$$
$$1_X \cdot (v - d_v) + \alpha \cdot \operatorname{div}\left(g(|\nabla u_\sigma|^2 + |\nabla v_\sigma|^2)\, \nabla v\right) = 0, \tag{3.17}$$

with $\nabla u_\sigma = \nabla(K_\sigma * u)$ and $\nabla v_\sigma = \nabla(K_\sigma * v)$ the gradients of the Gaussian smoothed flow components. For the function g one can for example choose the Charbonnier diffusivity (see [16])

$$g(s^2) = \frac{1}{\sqrt{1 + s^2/\lambda}}$$

or the Perona-Malik diffusivity (see [37])

$$g(s^2) = \frac{1}{1 + s^2/\lambda} \tag{3.18}$$

with $\lambda > 0$ a contrast parameter. Here we introduced a coupling between u and v via the joint diffusivity in order to prevent that discontinuities evolve at different location for the two flow components.

- Edge-enhancing Anisotropic Operator. We now make the step towards anisotropic operators that not only inhibit smoothing across motion boundaries, but further allow smoothing along them. Based on the superior results for image interpolation presented in [20] we consider *edge-enhancing anisotropic diffusion (EED)* for our vector approximation problem:

$$1_X \cdot (u - d_u) + \alpha \cdot \operatorname{div}\left(D(\nabla u_\sigma, \nabla v_\sigma)\, \nabla u\right) = 0, \tag{3.19}$$
$$1_X \cdot (v - d_v) + \alpha \cdot \operatorname{div}\left(D(\nabla u_\sigma, \nabla v_\sigma)\, \nabla v\right) = 0, \tag{3.20}$$

where $D(\nabla u_\sigma, \nabla v_\sigma)$ is a diffusion tensor. D should reflect the local flow structure and is chosen in such a way that it has the same set of eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ as the coupled tensor product

$$J_0 = \nabla u_\sigma \nabla u_\sigma^\top + \nabla v_\sigma \nabla v_\sigma^\top.$$

The eigenvalues λ_1 and λ_2 of D are chosen as follows:

- diffusion across motion edge: $\lambda_1 := g(\mu_1)$
- diffusion along motion edge: $\lambda_2 := 1$

where μ_1 is the eigenvalue of J_0 in the direction of the highest variation of the motion field and g is a diffusivity function as before. We can thus reduce the

diffusivity λ_1 perpendicular to motion edges with increasing variation μ_1, while we alow smoothing along the edges. The diffusion tensor D finally has the following form:

$$D := (\mathbf{v}_1\mathbf{v}_2) \cdot \mathrm{diag}(\lambda_1, \lambda_2) \cdot \begin{pmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \end{pmatrix}$$

The eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are in general not parallel to ∇u (to ∇v) or to $\nabla u^\top$ (to $\nabla v^\top$) and $D(\nabla u_\sigma, \nabla v_\sigma)$ therefore allows for anisotropic behavior.

- Flow-driven Anisotropic Operator. To obtain a more general type of anisotropic smoothing we extend the scalar valued function g from the isotropic case to a matrix valued function as follows: $g(A) := S \cdot \mathrm{diag}(g(\lambda_i)) \cdot S^\top$ where $A := S \cdot \mathrm{diag}(\lambda_i) \cdot S^\top$ is the eigenvalue decomposition of the symmetric matrix A. By applying g to both eigenvalues of J_0 as follows:

$$\lambda_1 := g(\mu_1),$$
$$\lambda_2 := g(\mu_2),$$

we can write the following *flow-driven anisotropic* approximation scheme:

$$1_X \cdot (u - d_u) + \alpha \cdot \mathrm{div}\left(g(\nabla u_\sigma \nabla u_\sigma^\top + \nabla v_\sigma \nabla v_\sigma^\top)\nabla u\right) = 0, \quad (3.21)$$
$$1_X \cdot (v - d_v) + \alpha \cdot \mathrm{div}\left(g(\nabla u_\sigma \nabla u_\sigma^\top + \nabla v_\sigma \nabla v_\sigma^\top)\nabla v\right) = 0, \quad (3.22)$$

with the diffusion tensor $D = g(\nabla u_\sigma \nabla u_\sigma^\top + \nabla v_\sigma \nabla v_\sigma^\top)$. Without any presmoothing of the flow components ($\sigma = 0$) one can even show for this type of operator that there exists a corresponding regularizer

$$\mathrm{tr}\ \Psi(\nabla u \nabla u^\top + \nabla v \nabla v^\top)$$

for the optic flow energy functional (see [43]). Here Ψ is a differentiable and increasing function for which $\Psi' = g$.

- Image-driven Anisotropic Operator. A valuable effect of anisotropic smoothing is the creation of sharp motion boundaries. Smoothing mainly happens within separate regions depending on the contrast parameter λ that is used in the diffusivity g. By tuning λ moving objects can be sharply distinguished within the data set. For flow-driven anisotropic operators though the motion boundaries can turn out to be irregular due to the limited spatial density of the data points. In general these boundaries will not accord to the real flow edges. If the motion boundaries of the flow field however coincide with object boundaries in the image we can try to overcome this problem by relying on *image-driven anisotropic* smoothing schemes. In our approach we base the diffusion tensor on the first image of the image pair:

$$1_X \cdot (u - d_u) + \alpha \cdot \mathrm{div}\left(g(J_\rho(\nabla f_{1\sigma}))\,\nabla u\right) = 0, \quad (3.23)$$
$$1_X \cdot (v - d_v) + \alpha \cdot \mathrm{div}\left(g(J_\rho(\nabla f_{1\sigma}))\,\nabla v\right) = 0, \quad (3.24)$$

where $J_\rho(\nabla f_{1\sigma})$ is the structure tensor of f_1. This approximation method will in some sense cluster the data according to structures in the underlying image. Smoothing of the flow field will be within image segments and as a consequence motion boundaries will be estimated with high accuracy if they coincide with the boundaries of the segments.

3.4.4 Post-smoothing of the Optic Flow Field

We have experienced that the approximation methods presented in 3.4.3 do not always give satisfactory results. In a region where the displacement data set F is very sparse, approximations with increasing smoothness weight $\alpha > 0$ can lead to bad fits. One can observe that the information from these isolated points does not disperse enough throughout the local neighborhood to give a good optic flow estimate there. We might call this an underfitting of the data in these sparse regions since increasing α corresponds to simplifying the fit.

To avoid this problem we keep to the following ad hoc approach. In the first step we fit our data using formula (3.11) with a smoothness weight $\alpha = 0$. The resulting data approximation is identical to pure interpolation. In the next step this initial flow estimation will be smoothed according to formula (3.7).

To formalize the second step we call f_u and f_v the u- and v-components of the optic flow estimate after the first step. We smooth this flow estimate according to:

$$c \cdot (u - f_u) - (1 - c) \cdot \operatorname{div}\left(\Psi'(|\nabla u|^2 + |\nabla v|^2)\,\nabla u\right) = 0, \tag{3.25}$$

$$c \cdot (v - f_v) - (1 - c) \cdot \operatorname{div}\left(\Psi'(|\nabla u|^2 + |\nabla v|^2)\,\nabla v\right) = 0, \tag{3.26}$$

where $0 < c < 1$ is a constant weight. As a smoothing operator we can use any of the afore mentioned operators. It would be advisable to use the same operator as in the interpolation step since the favorable effects of different operators can be easily undone by combining them.

The idea of post-smoothing can be advocated if we regard the inaccuracies in our initial flow estimate as high frequency noise introduced by interpolating our data set F. High frequency noise means that the fluctuations in the initial estimate are local and small compared to the meaningful flow structures in the exact displacement field. We can write equation (3.25) for the u-component with homogeneous smoothing where $(1 - c)/c$ is replaced by α as follows:

$$\frac{u - f_u}{\alpha} = \Delta u.$$

We can regard this as a fully implicit time discretization of the diffusion filter

$$\begin{aligned} \partial_t u &= \Delta u, \\ u(x, y, 0) &= f_u(x, y) \end{aligned}$$

with a single time step of size α. For homogeneous diffusion the diffusion time relates directly to the width σ of the Gaussian convolution kernel. By appropriately choosing the time step size α (the weight c) we can smooth out the small scale noise without blending the large scale flow structures. This is basically true for all other operators discussed so far.

3.5 Handling Outliers

So far we have assumed that the data set F is mainly corrupted by small-scale noise due to the limitations of our feature detector. It is very likely however that grossly erroneous displacements, so called outliers, are present and they can cause our function approximation schemes to fail. We would like to reserve the term outlier solely for displacement data $\mathbf{d}_i$ that result from spurious matches. Falsely matched features will usually give displacements that deviate from the true displacement in that location in both magnitude and direction.

An important feature of the SIFT algorithm with respect to outlying data is that the amount of outliers in the final data set can be partially controlled by the threshold of the distance ratio used in the matching step (see under 3.2.2). This is because of the high dimensionality of the descriptor space in which clear matches are inclined to cluster together. We have experienced that below a certain ratio threshold, just a small portion of the whole data set can be regarded as outliers. Raising the threshold results in supplementary feature correspondences of which many tend to be erroneous. The extra number of correct matches one can obtain by increasing the threshold might therefore not be worth the effort of sifting out the bad ones. This is especially true when no reliable methods for outlier removal are available.

The methods presented here serve essentially as examples to show that it is to a certain extent possible to deal with outliers. These methods are by no means optimal and better techniques surely exist. Because the emphasis of our work was not on conditioning our data set for subsequent use, we did not go through the trouble of optimizing the performance of the proposed methods.

There are generally two approaches to deal with the problem of outliers. One could try to *remove the outliers* from the data set prior to using the data, or one could use a *robust estimation* technique that makes use of the full data set but is not easily affected by outliers (see [38, 40]). Sometimes robustness is additionally interpreted as the ability of a method to deal with data sets drawn from different populations, as is the case with separately moving objects. We do not adopt this notion here. We have mentioned previously that data approximation based on anisotropic diffusion can sometimes provide us with segmentation like results that correspond to a clustering of different data populations.

We will first introduce a robust variant of our vector approximation scheme presented in 3.4.3. This robust method tries to remain insensitive to extreme displacement data by penalizing the residual error differently. After that we present a way of removing outliers from the data set before using it in the flow estimation.

A standard course to take in order to make a method like (3.2) more robust to outliers is to replace the squared error loss function by a subquadratic loss function. Many estimation techniques, such as M-estimators and L_1-regression, make use of this idea. Our formula for signal regularization (3.6) with nonquadratic penalization of the similarity term can be written as:

$$E(u) = \int_0^1 \left(c \cdot \Phi\left((u-f)^2\right) + (1-c) \cdot \Psi(u_x^2) \right) dx$$

where $\Phi(s^2)$ is an increasing function that is convex in s. The corresponding Euler-Lagrange equation is:

$$c \cdot \Phi'(u-f) - (1-c) \cdot \partial_x \left(\Psi'(u_x^2)\, u_x \right) = 0 \, .$$

From robust statistics we know that in the case of subquadratic penalization outliers will be penalized less severely than in the quadratic approach, but this will lead unavoidably to nonlinear methods. Suitable nonquadratic penalizers can be derived from nonlinear diffusion filter design such as total variation (TV) denoising. We opted for the TV-related L_1-penalizer which corresponds to the absolute error loss:

$$\Phi_{L_1}(s^2) = 2\sqrt{s^2 + \epsilon^2}$$

and

$$\Phi'_{L_1}(s^2) = \frac{1}{\sqrt{s^2 + \epsilon^2}}$$

where $\epsilon > 0$ denotes some fixed regularization parameter due to the non-differentiability of the L_1-norm. A robust approximation scheme for our displacement data with L_1-penalization of the similarity and e.g. homogeneous regularization can be derived from (3.12) and (3.13) as:

$$1_X \cdot \Phi'_{L_1}\left((u-d_u)^2+(v-d_v)^2\right)(u-d_u) - \alpha \cdot \Delta u \;=\; 0, \tag{3.27}$$

$$1_X \cdot \Phi'_{L_1}\left((u-d_u)^2+(v-d_v)^2\right)(v-d_v) - \alpha \cdot \Delta v \;=\; 0. \tag{3.28}$$

We have introduced a coupling between u and v in the robust term to obtain rotation invariance. This will in general give slightly better results.

Many methods have been established that determine a set of inliers from an original data set prior to performing a minimization. Many of them have proven their efficiency such as RANSAC and Least Median of Squares (LMS) (see [23]), but they are parametric in nature. If we want to exclude outliers from our displacement data we can only fall back on our initial assumption of smoothness. We are thus looking for methods that are able to filter out displacement data such that the resulting set F is predominantly smooth.

An interesting method for filtering discrete vector data is the so called *vector median filter (VMF)* presented by Astola *et al.* in [1]. Median filtering is a way of denoising a sampled signal by moving a window over the data points and replacing the value in the point by the median of the data within the window. Median filters are robust and suited for smoothing if the noise characteristic is not known. They are especially appropriate for removing salt and pepper noise (impulses) and they additionally preserve steps in the signal. For VMFs the definition of the median x_{med} of n scalar values x_i with $i = 1, ..., n$, given by

$$\sum_{i=1}^{n} |x_{\text{med}} - x_i| \leq \sum_{i=1}^{n} |y - x_i| \quad \text{for all } y$$

is extended to the vector median:

$$\sum_{i=1}^{n} |\mathbf{x}_{\text{med}} - \mathbf{x}_i|_p \leq \sum_{i=1}^{n} |\mathbf{x}_j - \mathbf{x}_i|_p \quad \text{for all } j \in \{1, ..., n\}.$$

The chosen norm $|\quad|_p$ can be either the L_1- or the L_2-norm. The vector median, as defined in this way, is always an element of the subset within the filter window but may not be unique. In case of non uniqueness a choice is usually made based on the spatial or temporal ordering of the data inside the window. In [1] the applicability of VMFs is properly demonstrated by smoothing a sampled velocity field.

The VMF has to be applied with caution to a scattered data set since the concept of the filter window might have to be revised. In our experiments we have implemented a method that is a modification of VMF and is sometimes used in tracking (see e.g. [17]). This method does not replace a data sample by a filtered output but discards a vector if it is not in accordance with the bulk of the neighboring vectors. We represent by $F' \subseteq F$ the set of displacements in the points $\{\mathbf{x}_k \,|\, k \in \{1, \ldots, n\}\}$ that are within a radius R of a point $\mathbf{x}_i$. We call $\mathbf{d}_c$ the element of F' that has the minimum median distance to all other elements $\mathbf{d}_k \in F'$ and it is chosen as the best representation of the subset F':

$$\underset{k \neq c}{\text{med}} \,|\, \mathbf{d}_c - \mathbf{d}_k \,|_2 \leq \underset{k \neq l}{\text{med}} \,|\, \mathbf{d}_l - \mathbf{d}_k \,|_2 \quad \text{for all } \mathbf{d}_l \in F'.$$

The displacement $\mathbf{d}_i$ in $\mathbf{x}_i$ is regarded as an outlier if it lies too far from $\mathbf{d}_c$. For this we apply a simple threshold of the distance: if

$$|\,\mathbf{d}_i - \mathbf{d}_c\,|_2 < c \cdot \operatorname*{med}_{k \neq c} |\,\mathbf{d}_c - \mathbf{d}_k\,|_2$$

with a parameter c, then $\mathbf{d}_i$ is an inlier, otherwise it is discarded. This rather heuristical method has two parameters: the radius of the neighborhood R and c. We can take R constant or make it dependent on the local point density to make sure that we always have the same amount of elements in F'. If R is too big we might not detect all outliers due to a too large variation in F'. If R is too small no variation is captured at al. We can give c a constant value, or a value depending on the distance distributions within the subset F'.

This approach for removing outliers from the data set F can be improved upon. The filter window of VMFs is replaced by a neighborhood of radius R but this might not solve all problems that arise from the fact that the data does not lie on a regular grid. Especially on the boundaries of regions with a different point density we expect that points in the sparser region near the boundary will be discarded. To overcome this we can think of duplicating data within the radius R wherever the data is sparse in order to obtain a balanced spatial distribution. This duplication should also be taken into account near the boundaries of the domain Ω. In our actual realization of the modified VMF we have not gone to the extend of implementing these ideas.

A final idea for removing outlying data is to make direct use of the optic flow constraint. We consider successful constancy assumptions from standard variational optic flow models such as a constancy of the grey value or the image gradient. Brightness constancy in a point $\mathbf{x}_i$ with displacement $\mathbf{d}_i$ means that

$$\mathbf{d}_i^\top \cdot \nabla_3 f(\mathbf{x}_i) = f_x(\mathbf{x}_i)\, d_{u_i} + f_y(\mathbf{x}_i)\, d_{v_i} + f_t \leq c \tag{3.29}$$

with c being a small threshold. The value of c is difficult to predict and is most likely dependent on the image pair used. Imposing constancy of the spatial brightness gradient $\nabla_2 f$ means

$$\mathbf{d}_i^\top \cdot \nabla_3 f_x(\mathbf{x}_i) + \mathbf{d}_i^\top \cdot \nabla_3 f_y(\mathbf{x}_i) \leq c. \tag{3.30}$$

To gain more accuracy we can think of using the non linearized constancy assumptions, like e.g. for brightness:

$$f(\mathbf{x}_i + \mathbf{d}_i) - f(\mathbf{x}_i) \leq c. \tag{3.31}$$

We can also combine different assumptions by imposing a threshold on their sum.

We can come up with other techniques, such as repetitively fitting a random subset of our data and discarding points that give too much variance for the fit in their location. This bootstrapping is computationally expensive and its performance will depend on the type of interpolant. One of the simplest techniques that has proven its usefulness however is *applying a threshold on the magnitude of the displacement*. This can only be done if we have knowledge about the maximum flow in the image sequence, but it will surely rule out grossly false matches.

Chapter 4

Combined Optic Flow Methods

Here we will return to the starting point of our work: to investigate to what extent a standard variational optic flow method can benefit from additional motion information in a discrete subset of the domain. Because of the symmetrical nature of the problem it is also interesting to consider in what way the vector approximation scheme from Chapter 3 can benefit from including an optic flow constraint. This chapter will present a combined optic flow method that can be either interpreted as an extension of the standard method or as an extension of the feature-based method.

We start from the idea of a convex combination of approximation and standard variational optic flow in the feature points X. Outside these points our objective is to have a standard optic flow estimation. For simplicity, both the smoothness operator for the approximation and the operator resulting from the regularizer in standard optic flow are taken to be the same. Furthermore let d_u and d_v denote the u- and v-component of the displacement field $\mathbf{d}$, let L denote some elliptic operator and let D_u and D_v denote the terms resulting from the data term in the standard optic flow method. If we use the brightness constancy assumption, then $D_u = f_x^2 u + f_x f_y v + f_x f_t$ and $D_v = f_x f_y u + f_y^2 v + f_y f_t$. Our general model can be written as:

$$\begin{aligned} c(\mathbf{x}) \cdot ((u - d_u) - \alpha \cdot Lu) + (1 - c(\mathbf{x})) \cdot (D_u - \beta \cdot Lu) &= 0, \qquad (4.1) \\ c(\mathbf{x}) \cdot ((v - d_v) - \alpha \cdot Lv) + (1 - c(\mathbf{x})) \cdot (D_v - \beta \cdot Lv) &= 0, \qquad (4.2) \end{aligned}$$

where α and β are smoothness weights and

$$c(\mathbf{x}) = \begin{cases} \in [0,1] & \text{if } \mathbf{x} \in \{\mathbf{x}_0, ..., \mathbf{x}_n\} \\ 0 & \text{else} \end{cases} \qquad (4.3)$$

a confidence function.

To improve our understanding of this model we write equations (4.1) and (4.2) separately on Ω and $\Omega \setminus X$. We get the following:

$$\left.\begin{array}{rcl} c\cdot(u-d_u)+(1-c)\cdot D_u- & & \\ c\cdot\alpha\cdot Lu-(1-c)\cdot\beta\cdot Lu & = & 0 \\ c\cdot(v-d_v)+(1-c)\cdot D_v- & & \\ c\cdot\alpha\cdot Lv-(1-c)\cdot\beta\cdot Lv & = & 0 \end{array}\right\} \quad \text{if } \mathbf{x}\in\{\mathbf{x}_0,...,\mathbf{x}_n\}$$

$$\left.\begin{array}{rcl} D_u-\beta\cdot Lu & = & 0 \\ D_v-\beta\cdot Lv & = & 0 \end{array}\right\} \quad \text{else} \tag{4.4}$$

where $c\in[0,1]$. These equations can be interpreted in three ways:

- ◇ We can regard our standard optic flow method as being augmented by including information of feature displacements in a finite number of points X. The parameter c controls how much significance we want to give to this information. If $c=0$ the equations result in a conventional optic flow estimation as introduced in Chapter 2. If on the other hand $c=1$, the full displacement values $\mathbf{d}_i$ are assigned to the positions $\mathbf{x}_i$, $i=0,\ldots,n$, potentially regularized by a smoothness imposed by α.

- ◇ On the other hand we can state that our approximation scheme from Chapter 3 has been extended by adding the terms D_u and D_v from the optic flow constraint. We can regulate the influence of these terms through c. We point out that we approach the original approximation scheme for $c\to 1$ and $\beta\to\infty$.

- ◇ In conclusion we can look upon these equations at a high level as a convex combination of a standard variational model and a feature-based model for optic flow calculation:

$$\tilde{c}\cdot[\,1_X\cdot(u-d_u)-\tilde{\alpha}\cdot Lu\,]+(1-\tilde{c})\cdot\Big[\,D_u-\tilde{\beta}\cdot Lu\Big] \;=\; 0, \tag{4.5}$$

$$\tilde{c}\cdot[\,1_X\cdot(v-d_v)-\tilde{\alpha}\cdot Lv\,]+(1-\tilde{c})\cdot\Big[\,D_v-\tilde{\beta}\cdot Lv\Big] \;=\; 0, \tag{4.6}$$

  where $\tilde{\alpha}$ and $\tilde{\beta}$ are smoothness weights and $0<\tilde{c}<1$ is a constant over the whole domain Ω. Although the operator L in the notation given above is split between the two models, its influence is global. It should be noted that the smoothness weight $\tilde{\alpha}$ of the feature-based model can take on any value outside X and is thus not clearly defined on $\Omega\setminus X$. The equations of this generic model should not serve as the basis for an implementation.

Our model (4.4) is complicated because of the three independent parameters α, β and c. We can reduce the degrees of freedom by considering the special case for $\alpha=0$. Our model will be:

$$\begin{aligned} c(\mathbf{x})\cdot\Phi'\left((u-d_u)^2+(v-d_v)^2\right)\cdot(u-d_u)+& \\ (1-c(\mathbf{x}))\cdot(D_u-\beta\cdot Lu) \;&=\; 0, \end{aligned} \tag{4.7}$$

$$\begin{aligned} c(\mathbf{x})\cdot\Phi'\left((u-d_u)^2+(v-d_v)^2\right)\cdot(v-d_v)+& \\ (1-c(\mathbf{x}))\cdot(D_v-\beta\cdot Lv) \;&=\; 0, \end{aligned} \tag{4.8}$$

with $c(\mathbf{x})$ defined as in expression (4.3) and with a nonquadratic penalizer $\Phi(s^2)$ such as the L_1-penalizer $\Phi_{L_1}(s^2)$ introduced earlier in section 3.5. The penalization grants some degree of robustness against feature displacements that have large deviations from the standard optic flow estimation. If $c(\mathbf{x})$ is binary, this set of equations can be thought of as "pinning down" the conventional optic flow estimation in the points $\mathbf{x}_i$ to $\mathbf{d}_i$, $i = 0, \ldots, n$. Equivalently, we can claim to be using a conventional optic flow model as interpolant of a set of scattered motion vectors. $c(\mathbf{x}) \in [0, 1]$ in the data locations $\mathbf{x}_i$ allows for the possibility to scale down the importance of the feature displacements due to extra smoothness and data constraints.

Another idea is to evaluate the similarity measure against the optic flow constraint:

$$\begin{aligned} c(\mathbf{x}) \cdot \Phi' \left((u - d_u)^2 + (v - d_v)^2 \right) \cdot (u - d_u) + & \\ (1 - c(\mathbf{x})) \cdot D_u - \alpha \cdot Lu &= 0, \qquad (4.9) \\ c(\mathbf{x}) \cdot \Phi' \left((u - d_u)^2 + (v - d_v)^2 \right) \cdot (v - d_v) + & \\ (1 - c(\mathbf{x})) \cdot D_v - \alpha \cdot Lv &= 0. \qquad (4.10) \end{aligned}$$

This special case arises from model (4.4) for $\alpha = \beta$. In this case the smoothness makes "jumps" relative to the terms D_u and D_v. This model might be considered unrealistic due to the constant smoothness weight α. D_u and D_v have a different order of magnitude than the similarity terms. They will probably require other values for α than the ones that are optimal in conjunction with the similarity terms.

Because the various terms in the equations presented here work at different scales, it is difficult to predict a value for any of the parameters in order to obtain an expected result. When implementing one of the previous schemes for experimentation one has to allow enough numerical precision for the parameters.

Chapter 5

Discretization and Algorithmic Realization

Up till now we have studied several models for optic flow calculation in a continuous setting. In all cases the unknown flow field is the solution of a system of two coupled PDEs. To solve for the unknown flow components we have to discretize these PDEs by approximating the derivatives by finite differences. The result of this discretization is in general a nonlinear system of equations that has to be solved by numerical algorithms. Its solution is an approximation of the true optic flow field for a finite number of points within the integration domain. Since we are working with digital images, the discretization is imposed in a natural way by the pixel grid which makes up the image.

The discretization of the PDEs is the topic of the first section of this chapter. We cover various schemes based on the smoothing operator that appears in the PDE. The discretizations of the data constraint and the similarity constraint are presented together with the discretization of the homogeneous operator. For the remaining smoothing operators that have been discussed previously we propose different stencil notations. In the last section we study the resulting nonlinear system of equations and its solution.

5.1 Discretization of the PDEs

In the following we assume that the integration domain Ω is discretized by a grid of $N = n_x \times n_y$ pixels such that a pixel (i, j) with $1 \leq i \leq n_x$ and $1 \leq j \leq n_y$ represents the location (x_i, y_i) where

$$\begin{aligned} x_i &= (i - \tfrac{1}{2})h_x, \\ y_j &= (j - \tfrac{1}{2})h_y, \end{aligned}$$

We further assume that the grid is equidistant with a grid size $h_x = h_y = h$. We then denote by $u_{i,j}$ and $v_{i,j}$ the approximations of u and v in the location (x_i, y_j).

5.1.1 The Homogeneous Operator

A finite difference approximation of the Euler-Lagrange equations (2.10)-(2.11) of the Horn and Schunck method is given by

$$J_{11\,i,j}u_{i,j} + J_{12\,i,j}v_{i,j} + J_{13\,i,j} - \alpha \cdot \sum_{\mathcal{N}(i,j)} \frac{u_{k,l} - u_{i,j}}{h^2} = 0, \tag{5.1}$$

$$J_{12\,i,j}u_{i,j} + J_{22\,i,j}v_{i,j} + J_{23\,i,j} - \alpha \cdot \sum_{\mathcal{N}(i,j)} \frac{v_{k,l} - v_{i,j}}{h^2} = 0, \tag{5.2}$$

for $i = 1, ..., n_x$ and $j = 1, ..., n_y$. Here we denote by $J_{mn\,i,j}$ the component (m, n) of $J_0(\nabla_3 f)$ in some pixel (i, j) and by $\mathcal{N}(i, j)$ a set of neighbors of pixel (i, j). In the case of the discretized Laplacian the summation is performed over the four neighbors $(k, l) \in \mathcal{N}(i, j)$ in x- and y-direction. The same finite difference approximation of the equations (3.27) and (3.28) is given by

$$1_X \cdot \Phi'_{i,j}\, u_{i,j} - 1_X \cdot \Phi'_{i,j}\, d_{u\,i,j} - \alpha \cdot \sum_{\mathcal{N}(i,j)} \frac{u_{k,l} - u_{i,j}}{h^2} = 0, \tag{5.3}$$

$$1_X \cdot \Phi'_{i,j}\, v_{i,j} - 1_X \cdot \Phi'_{i,j}\, d_{v\,i,j} - \alpha \cdot \sum_{\mathcal{N}(i,j)} \frac{v_{k,l} - v_{i,j}}{h^2} = 0, \tag{5.4}$$

where $d_{u\,i,j}$ and $d_{v\,i,j}$ are the u- and v-components of $\mathbf{d}$ in pixel (i, j). Note that these components are only known for pixel locations where a feature correspondence $\mathbf{x}_q \longleftrightarrow \mathbf{x}'_q$ has been established. $\Phi'_{i,j}$ is the derivative of some penalizing function $\Phi\left((u - d_u)^2 + (v - d_v)^2\right)$ in pixel (i, j).

Gaussian convolution is realized by discrete convolution with a renormalized Gaussian that is truncated at three times the standard deviation. Spatial derivatives of the image in the motion tensor entries are approximated using the fourth-order stencil $(-1, 8, 0, -8, 1)/(12h)$. Temporal derivatives are approximated with a simple two-point stencil.

If we take h to be 1, the above discretization of the Laplacian corresponds to the simple second order stencil

	1	
1	−4	1
	1	

. (5.5)

The homogeneous Neumann boundary conditions can be implemented by mirroring the boundary pixel values and thereby creating a halo region with a width of one pixel. Another way of handling the Neumann boundary conditions is to use an adaptive stencil that depends on whether the pixel lies inside the domain or on the boundary. Stencil (5.5) would for instance take the forms

1	−3	1
	1	

and

−2	1
1	

in an upper boundary pixel and an upper corner, respectively. An adaptive stencil can be precalculated for every pixel location. This avoids mirroring updated values in every step of the iterative solution. If we use a boundary layer in which the pixel values are set to 0, we only need to precalculate the central stencil weight which can be either -2, -3 or -4. When using the adaptive stencil approach we consider the number of neighboring pixels in $\mathcal{N}(i, j)$ to be location dependent.

5.1.2 The Biharmonic Operator

A stencil for the biharmonic operator Δ^2 can be obtained by convolving the stencil of the Laplacian (5.5) with itself. For an inner pixel this (5×5) stencil is given by:

		1		
	2	−8	2	
1	−8	20	−8	1
	2	−8	2	
		1		

. (5.6)

For Neumann boundary conditions the stencil looks different for boundary pixels and corner pixels. Therefore we developed a stencil notation that is a function of the pixel location and makes use of the indicator function. The indicator function of the set $\{i \,|\, i > 1\}$ for instance will be denoted in short as

$$1_{(i>1)} = \begin{cases} 1 & \text{if } i > 1 \\ 0 & \text{if } i \leq 1 \end{cases}$$

and can simply be implemented as a test. Additionally we denote by $S_{i,j}$ the central weight of the Laplacian stencil in the pixel location (i, j). We can then write the biharmonic stencil in a general form as:

		$1_{(j<n_y-1)}$		
	$2 \cdot (1_{(i>1)} \cdot 1_{(j<n_y)})$	$1_{(j<n_y)} \cdot (S_{i,j} + S_{i,j+1})$	$2 \cdot (1_{(i<n_x)} \cdot 1_{(j<n_y)})$	
$1_{(i>2)}$	$1_{(i>1)} \cdot (S_{i,j} + S_{i-1,j})$	$S_{i,j}^2 + 1_{(i>1)} + 1_{(i<n_x)} + 1_{(j>1)} + 1_{(j<n_y)}$	$1_{(i<n_x)} \cdot (S_{i,j} + S_{i+1,j})$	$1_{(i<n_x-1)}$
	$2 \cdot (1_{(i>1)} \cdot 1_{(j>1)})$	$1_{(j>1)} \cdot (S_{i,j} + S_{i,j-1})$	$2 \cdot (1_{(i<n_x)} \cdot 1_{(j>1)})$	
		$1_{(j>2)}$		

.

Using this expression the stencil can be precalculated for every pixel. By introducing a boundary layer with a width of two pixels in which the values are set to 0, we do not need to perform all the tests. We only need to precalculate the central stencil weight and the four weights on the diagonals. Because we combined the homogeneous and the biharmonic smoothing operators in our experiments, the weights $S_{i,j}$ were available for every pixel. Alternatively we can mirror the two layers of pixel values closest to the boundaries in every iteration step to create a boundary layer. This allows us to use the full stencil (5.6).

5.1.3 Nonlinear Operators

We will use the general notation $\operatorname{div}(D \nabla u)$ and $\operatorname{div}(D \nabla v)$ for any nonlinear operator that has been discussed so far. The diffusion tensor D will be a function of u and v opposed to the linear case where D only depends on x and y. Note that this notation also covers the isotropic operator where $D = g(|\nabla u_\sigma|^2 + |\nabla v_\sigma|^2)\, I$, where I

is the 2 by 2 identity matrix. In what follows we will illustrate the discretization of the nonlinear operator for the flow component u. The discretization of $\mathrm{div}(D\,\nabla v)$ proceeds in the same way.

If we write $D = \left(\begin{smallmatrix} a & b \\ b & c \end{smallmatrix}\right)$ then

$$\begin{aligned} \mathrm{div}(D\,\nabla u) &= \mathrm{div}\begin{pmatrix} a\,\partial_x u + b\,\partial_y u \\ b\,\partial_x u + c\,\partial_y u \end{pmatrix} \\ &= \partial_x(a\,\partial_x u) + \partial_x(b\,\partial_y u) + \partial_y(b\,\partial_x u) + \partial_y(c\,\partial_y u). \end{aligned}$$

The terms $\partial_x(a\,\partial_x u)$ and $\partial_y(c\,\partial_y u)$ are being discretized by using backward and forward finite differences (see [41]):

$$\begin{aligned} \partial_x(a\,\partial_x u) &= \frac{1}{h}\left(\frac{a_{i+1,j}+a_{i,j}}{2}\,\frac{u_{i+1,j}+u_{i,j}}{h} - \frac{a_{i,j}+a_{i-1,j}}{2}\,\frac{u_{i,j}+u_{i-1,j}}{h}\right), \\ \partial_y(c\,\partial_y u) &= \frac{1}{h}\left(\frac{c_{i,j+1}+c_{i,j}}{2}\,\frac{u_{i,j+1}+u_{i,j}}{h} - \frac{c_{i,j}+c_{i,j-1}}{2}\,\frac{u_{i,j}+u_{i,j-1}}{h}\right), \end{aligned}$$

where we approximate the entries of D in a pixel (i,j) by $a_{i,j}$, $b_{i,j}$ and $c_{i,j}$.

For isotropic diffusion we have $b = 0$ and $a = c = g(\nabla u_\sigma, \nabla v_\sigma)$. In this case the above discretization is sufficient. It results in a (3×3) stencil which is similar to the Laplacian stencil (5.5) but with non-constant entries. In the case of anisotropic diffusion we have to take the mixed terms $\partial_x(b\,\partial_y u)$ and $\partial_y(b\,\partial_x u)$ into account. If we apply a standard approximation by central differences to the mixed terms, we extend our (3×3) stencil with 4 weights on the stencil diagonals. These weights can have an arbitrary sign since b can be positive or negative. As a consequence, the requirement that the off-diagonal entries of the resulting system matrix have to be nonnegative may be violated. This non-negativity requirement assures so called stability in the maximum norm which means that the discretization satisfies a maximum-minimum principle. To ensure a stable discretization we use the (3×3) stencil proposed in [41] and depicted in Figure 5.1. This second order discretization only guarantees nonnegativity if the condition number of D is smaller than a certain value. It is however expected that over- and undershoots remain bounded if this criterion is not met. The same stencil is also used for the linear image-driven anisotropic operator.

5.2 Solution of the System of Equations

By adopting a certain ordering (e.g. row major ordering) we can arrange the $2N$ unknowns of the discretized PDEs in a single vector $\left(\begin{smallmatrix} \mathbf{u} \\ \mathbf{v} \end{smallmatrix}\right)$. The column vector $\mathbf{u}$ consists of the values $u_{i,j}$ and the column vector $\mathbf{v}$ of the values $v_{i,j}$ for $i = 1, ..., n_x$ and $j = 1, ..., n_y$. All the discretizations discussed in the previous section will lead to a system of equations with a sparse $2N \times 2N$ system matrix. This system matrix consists of two parts: a part that results from the data or similarity constraint and a part that results from the smoothness constraint. The system of equations can be written in a general form as (see [9])

$$\left(\begin{pmatrix} A_{11} & A_{12} \\ A_{12} & A_{22} \end{pmatrix} - \alpha \cdot \begin{pmatrix} B & 0 \\ 0 & B \end{pmatrix}\right)\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix}.$$

$\frac{\lvert b_{i-1,j+1}\rvert - b_{i-1,j+1}}{4h_1h_2}$ $+\frac{\lvert b_{i,j}\rvert - b_{i,j}}{4h_1h_2}$	$\frac{c_{i,j+1}+c_{i,j}}{2h_2^2} - \frac{\lvert b_{i,j+1}\rvert + \lvert b_{i,j}\rvert}{2h_1h_2}$	$\frac{\lvert b_{i+1,j+1}\rvert + b_{i+1,j+1}}{4h_1h_2}$ $+\frac{\lvert b_{i,j}\rvert + b_{i,j}}{4h_1h_2}$
$\frac{a_{i-1,j}+a_{i,j}}{2h_1^2}$ $-\frac{\lvert b_{i-1,j}\rvert + \lvert b_{i,j}\rvert}{2h_1h_2}$	$-\frac{a_{i-1,j}+2a_{i,j}+a_{i+1,j}}{2h_1^2}$ $-\frac{\lvert b_{i-1,j+1}\rvert - b_{i-1,j+1} + \lvert b_{i+1,j+1}\rvert + b_{i+1,j+1}}{4h_1h_2}$ $-\frac{\lvert b_{i-1,j-1}\rvert + b_{i-1,j-1} + \lvert b_{i+1,j-1}\rvert - b_{i+1,j-1}}{4h_1h_2}$ $+\frac{\lvert b_{i-1,j}\rvert + \lvert b_{i+1,j}\rvert + \lvert b_{i,j-1}\rvert + \lvert b_{i,j+1}\rvert + 2\lvert b_{i,j}\rvert}{2h_1h_2}$ $-\frac{c_{i,j-1}+2c_{i,j}+c_{i,j+1}}{2h_2^2}$	$\frac{a_{i+1,j}+a_{i,j}}{2h_1^2}$ $-\frac{\lvert b_{i+1,j}\rvert + \lvert b_{i,j}\rvert}{2h_1h_2}$
$\frac{\lvert b_{i-1,j-1}\rvert + b_{i-1,j-1}}{4h_1h_2}$ $+\frac{\lvert b_{i,j}\rvert + b_{i,j}}{4h_1h_2}$	$\frac{c_{i,j-1}+c_{i,j}}{2h_2^2} - \frac{\lvert b_{i,j-1}\rvert + \lvert b_{i,j}\rvert}{2h_1h_2}$	$\frac{\lvert b_{i+1,j-1}\rvert - b_{i+1,j-1}}{4h_1h_2}$ $+\frac{\lvert b_{i,j}\rvert - b_{i,j}}{4h_1h_2}$

Figure 5.1: *A nonnegative second order discretization for* $\operatorname{div}(D\,\nabla u)$.

The submatrices A_{mn}, for $m, n \in \{1, 2\}$ result from the data and similarity constraints. For the brightness constancy assumption they consist of the values $J_{11i,j}$, $J_{22i,j}$ and $J_{12i,j}$ which introduce coupling between the two flow components. For a similarity constraint there is only coupling between the flow components if we use the robust formulation. Due to the dependency of $\Phi'_{i,j}$ on $u_{i,j}$ and $v_{i,j}$ there will also be an additional nonlinearity.

The weights in the stencils presented earlier appear in the row of the submatrix B that corresponds to the actual pixel. The center weight stands on the main diagonal while the remaining weights induce a block diagonal structure. Nonlinear operators introduce coupling between the two flow components via this part of the system matrix.

The right-hand side of the system gathers the values $J_{13i,j}$ and $J_{23i,j}$ or the known displacement components $d_{ui,j}$ and $d_{vi,j}$. Also here nonlinearities arise if we make use of any robust formulation. For a more detailed analysis of the resulting system of equations the reader is referred to [9].

To deal with the nonlinearities an outer fixed point iteration is applied. Within every outer iteration step Φ' and D are updated using the vector $\binom{\mathbf{u}}{\mathbf{v}}$ from the previous iteration and they are kept fixed. In this way we are allowed to solve a linear system of equations within every outer iteration step. This sparse linear system has a symmetric positive semidefinite system matrix which is diagonal dominant. It is effectively solved by *successive overrelaxation (SOR)* (see [46]) which is a variant of the Gauss-Seidel relaxation method. If the upper index denotes the iteration step and h is taken to be 1, the SOR method for the discretization (5.1)-(5.2) can be written as

$$u_{i,j}^{t+1} = (1-\omega)u_{i,j}^{t} + \omega \frac{-J_{12i,j}v_{i,j}^{t} - J_{13i,j} - \alpha \cdot \left(\sum_{\mathcal{N}^-(i,j)} u_{k,l}^{t+1} + \sum_{\mathcal{N}^+(i,j)} u_{k,l}^{t} \right)}{J_{11i,j} - \alpha \cdot S_{i,j}},$$

$$v_{i,j}^{t+1} = (1-\omega)v_{i,j}^{t} + \omega \frac{-J_{12i,j}u_{i,j}^{t} - J_{23i,j} - \alpha \cdot \left(\sum_{\mathcal{N}^-(i,j)} v_{k,l}^{t+1} + \sum_{\mathcal{N}^+(i,j)} v_{k,l}^{t} \right)}{J_{22i,j} - \alpha \cdot S_{i,j}},$$

where

$$\begin{aligned} \mathcal{N}^-(i,j) &= \{(k,l) \in \mathcal{N}(i,j) \,|\, k < i, l < j\}, \\ \mathcal{N}^+(i,j) &= \{(k,l) \in \mathcal{N}(i,j) \,|\, k > i, l > j\}, \end{aligned}$$

and $S_{i,j}$ denotes the center weight of the Laplacian stencil. Analogously the SOR method for (5.3)-(5.4) can be written as

$$u_{i,j}^{t+1} = (1-\omega)u_{i,j}^{t} + \omega \frac{1_X \cdot \Phi'_{i,j}\, d_{u\,i,j} - \alpha \cdot \left(\sum_{\mathcal{N}^-(i,j)} u_{k,l}^{t+1} + \sum_{\mathcal{N}^+(i,j)} u_{k,l}^{t} \right)}{1_X \cdot \Phi'_{i,j} - \alpha \cdot S_{i,j}},$$

$$v_{i,j}^{t+1} = (1-\omega)v_{i,j}^{t} + \omega \frac{1_X \cdot \Phi'_{i,j}\, d_{v\,i,j} - \alpha \cdot \left(\sum_{\mathcal{N}^-(i,j)} v_{k,l}^{t+1} + \sum_{\mathcal{N}^+(i,j)} v_{k,l}^{t} \right)}{1_X \cdot \Phi'_{i,j} - \alpha \cdot S_{i,j}}.$$

We can easily adapt this notation to any other stencil by replacing $S_{i,j}$ by the respective center weight and by multiplying the terms under the two summations by the stencil weights of the according position in $\mathcal{N}(i,j)$. The relaxation parameter ω influences the convergence speed and is usually chosen between 1.9 and 1.99 for optimal performance. Since the SOR method is globally convergent we initialize the flow components by 0. After a fixed number of SOR steps the updated flow components are used to recalculate Φ' and D in the next outer iteration.

Chapter 6

Experimental Results

We will present an experimental comparison of the performance of the three methods for optic flow estimation that have been previously discussed. These are the standard optic flow method with a data and a smoothness constraint (see Chapter 2), the feature-based approximation method that uses prescribed displacements in a number of domain points (see Chapter 3), and the combined optic flow method that results from adding a similarity term to the differential equations of the standard optic flow method (see Chapter 4). The comparison of these three methods will include an assessment of different smoothing operators. While the classical optic flow estimation benefits from optimizing the data term as well as the regularizer, our approximation scheme is only sensitive to alternative choices of the smoothing operator. As data term we will only consider the brightness constancy assumption as introduced in the method of Horn and Schunck.

The first part of this chapter shortly discusses implementation issues, error measures and test sequences used in our experiments. After that we describe a preliminary test run that serves as a motivation for the attempt of incorporating a feature detector into optic flow estimation. By including ground truth information in a simple variational optic flow method we get an indication of the potential improvement that we can obtain. In the remaining tests we use the SIFT algorithm as the actual feature detector. It will provide us with a set of local displacements that will be evaluated against a simple variational optic flow method. Early results will lead us to the concept of robust approximation. After introducing this important paradigm we assess the performance of the three optic flow methods referred to above. In a final series of experiments we try to select a high quality subset from our SIFT displacements using the strategies from 3.5.

6.1 Some Words on the Implementation

The test program was written in ANSI C and built around a routine performing SOR to solve the linear system of equations that results from the discretization of the PDEs (see Chapter 5). The program includes modules for memory allocation, input and output of various formats (.pgm, .ppm, binary and text data) and the visualization of results in OpenGL.

10 SOR iterations make up the inner loop of a fixed point iteration scheme. In the outer loop a potential periodic update of any robust terms or coefficients due to nonlinear operators can be carried out. To keep track of the position of the feature points used in the estimation a *mask* is generated with the same dimensions as the image data. In this mask the pixel location of each feature is highlighted and it serves as an input of the SOR routine. Wherever a mask entry has been set,

the according displacement can be utilized as a preset flow vector. The maximum number of mask entries that can be set is the number of pixels for which ground truth is available.

The code for generating SIFT features from an image is provided as an executable by David Lowe on his web site `http://www.cs.ubc.ca/~lowe/`. This code writes all SIFT features found in an image, referred to as key points, in a formatted text file. This file gives for every key the row and the column location, the scale, the orientation and the descriptor vector. We have integrated in our program the supplied code for reading in the key files of two images and matching the features. Matching a feature is done by an exhaustive search for the closest neighbor in the other key file. We can afford such an expensive operation because we are not interested in time efficiency. The resulting matches are stored as a list of structures. Each structure contains two SIFT features that form a pair.

The location of a SIFT feature is determined to sub-pixel precision. The value of the corresponding feature displacement should therefore be distributed over the neighboring pixels. To avoid too much complication due to extrapolation we assign the displacement to the nearest rounded pixel location. We do not believe that this has large effects on our results. In some cases, as for instance in the implementation of formula (3.29), or for the calculation of ground truth in a feature location, we made use of bilinear interpolation.

6.2 Assessment and Error Measures

As a principal measure for the quality of the optic flow estimation we use the widely accepted average angular error (AAE) (see [2]) defined as:

$$\mathrm{AAE}\,(\mathbf{w}_e, \mathbf{w}_c) = \frac{1}{|\Omega|} \int_\Omega \arccos\left(\frac{\mathbf{w}_e^\top \mathbf{w}_c}{|\mathbf{w}_e||\mathbf{w}_c|}\right) d\mathbf{x}, \tag{6.1}$$

with $|\Omega| = \int_\Omega d\mathbf{x}$ and

$$\arccos\left(\frac{\mathbf{w}_e^\top \mathbf{w}_c}{|\mathbf{w}_e||\mathbf{w}_c|}\right) = \arccos\left(\frac{u_e u_c + v_e v_c + 1}{\sqrt{(u_e^2 + v_e^2 + 1)(u_c^2 + v_c^2 + 1)}}\right),$$

the angular error between the estimated flow field $\mathbf{w}_e = (u_e, v_e, 1)$ and the correct flow field $\mathbf{w}_c = (u_c, v_c, 1)$. The AAE is calculated in the spatiotemporal domain and takes into account the arbitrary value of the time component. Alternatively we can assess our approximation schemes with an error measure of the form $\frac{1}{\Omega}\int_\Omega \mathcal{L}(\mathbf{w}_e, \mathbf{w}_c) d\mathbf{x}$ motivated by the loss function $\mathcal{L}$, as is usual in regression analysis (see [24]). We only mention the average squared error:

$$\begin{aligned} \mathrm{ASE}\,(\mathbf{w}_e, \mathbf{w}_c) &= \frac{1}{|\Omega|} \int_\Omega |(u_e, v_e)^\top - (u_c, v_c)^\top|_2^2 \, d\mathbf{x} \\ &= \frac{1}{|\Omega|} \int_\Omega \left((u_e - u_c)^2 + (v_e - v_c)^2\right) d\mathbf{x}. \end{aligned} \tag{6.2}$$

In this error measure only the spatial magnitude of the flow field is taken into account and it might be considered to capture the quality of the flow estimation better than the AAE. It can theoretically lead to unbounded errors and will not be used much in literature.

For the visualization and the qualitative evaluation of the computed flow field we use the color representation shown in Figure 6.1. The color indicates the direction of the displacements and the brightness expresses their magnitude.

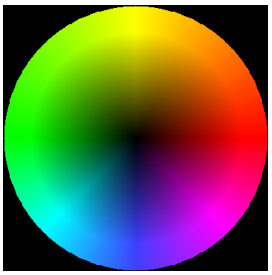

Figure 6.1: *The color code: the color indicates the direction of the displacement field and the brightness the magnitude.*

6.3 The Test Sequences

The image sequences that have been used in the experiments are well known test sequences for which the ground truth flow is known. The four sequences feature various types of motion, both smooth and discontinuous. They are the following:

- Yosemite sequence without clouds (Figure 6.2). This is a synthetic sequence of a flight over Yosemite Valley. Divergent motion dominates the mountain part with a maximum of 5 pixels per frame in the lower left corner. The total sequence consists of 15 frames and between frame 8 and 9 a ground truth displacement is available. The region above the horizon where no information on the ground truth is available is not included in the calculation of the AAE. The dimension is 316×252.

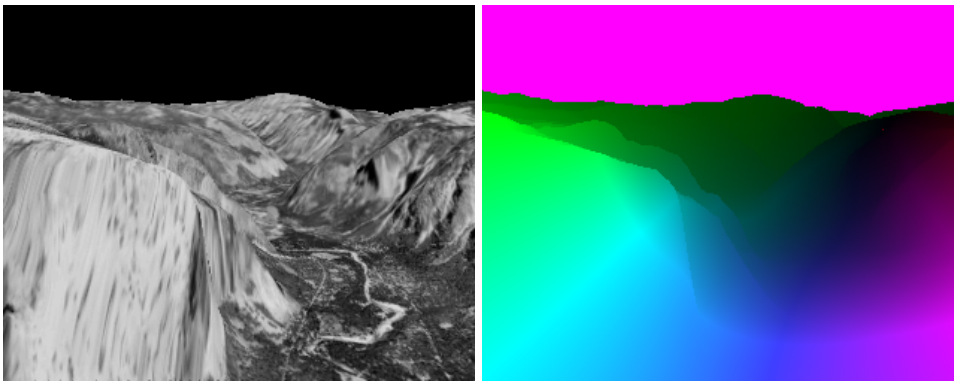

Figure 6.2: *The Yosemite sequence without clouds.* **Left:** *Frame 8 of the sequence.* **Right:** *The ground truth for the optic flow field between frames 8 and 9. The sky region is colored purple because the flow components have been set to -100 which indicates the absence of ground truth.*

- Yosemite sequence with clouds (Figure 6.3). This sequence is identical to the previous one but it has dark moving clouds as a backdrop. The clouds translate to the right with a speed of about 2 pixels per frame while slowly changing shape. They introduce varying illumination and do not provide texture. This makes it difficult for most methods to estimate the optic flow field. Additionally a motion discontinuity is created at the horizon. The sequence can be downloaded from `ftp://ftp.csd.uwo.ca/pub/vision`.

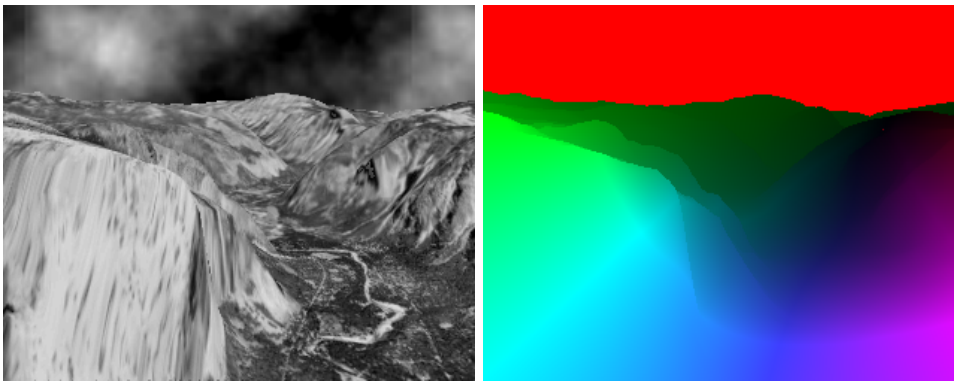

Figure 6.3: *The Yosemite sequence with clouds.* **Left:** *Frame 8 of the sequence.* **Right:** *The ground truth for the optic flow field between frames 8 and 9.*

- Old Marble sequence (Figure 6.4). This is a real world sequence showing four dark marbled blocks that move along with the marble floor to the lower right. The white marble block in the center stands still. There are clear discontinuities in the flow field but they do not involve radical switches in direction. The sequence consists of 32 frames with a ground truth for frames 16 and 17. The displacements are up to 3 pixels and the ground truth is not given for the upper part of the background and the edges of the marble blocks. The dimension is 512×512. The sequence can be downloaded from `http://i21www.ira.uka.de/image_sequences`.

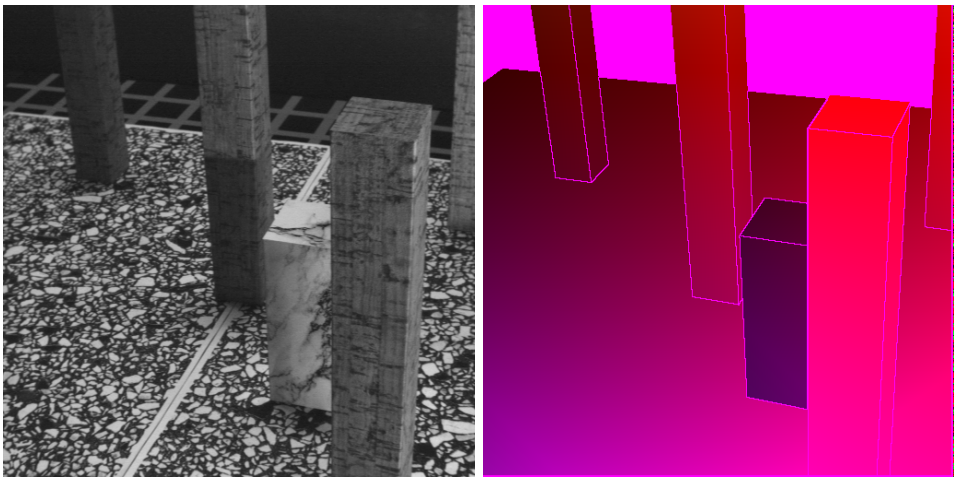

Figure 6.4: *The old Marble sequence.* **Left:** *Frame 16 of the sequence.* **Right:** *The ground truth for the optic flow field between frames 16 and 17. The background is colored purple because the flow components have been set to -100 which indicates the absence of ground truth.*

- New Marble sequence (Figure 6.5). This synthetic sequence shows two translating marble blocks and a static one on a heavily textured static surface. There are sharp motion discontinuities that coincide with the object boundaries. The sequence is 200 frames long and for frames 150 and 151 a ground truth is available. The dimension is 512×384. This sequence is available from `http://i21www.ira.uk.de/image_sequences`.

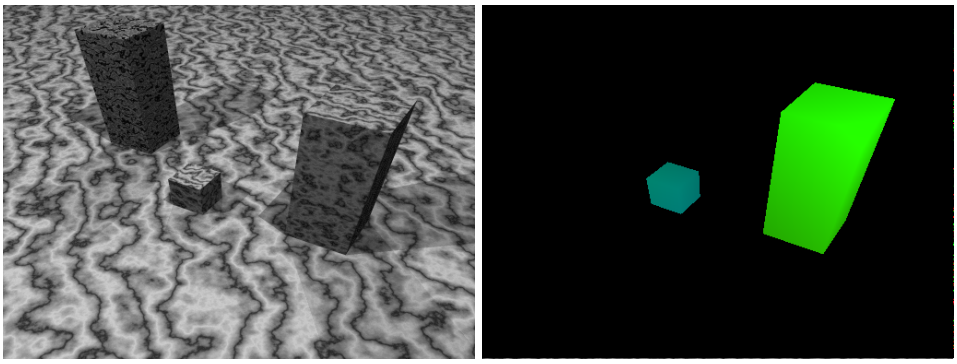

Figure 6.5: *The new Marble sequence.* **Left:** *Frame 150 of the sequence.* **Right:** *The ground truth for the optic flow field between frames 150 and 151.*

6.4 Experiments with Ground Truth

In a preliminary series of experiments we try to get an idea to what extent the use of discrete point information can ameliorate a variational optic flow estimation. Additionally, we wish to derive basic conditions which the discrete subset of points has to fulfill in order to give satisfactory improvements. To this end we extract a number of pixel locations and we assign to them the ground truth of that location. We survey the improvement in flow calculation obtained for a certain number of preset displacements. Our points of interest are selected in the following ways:

i. uniformly at random over the image domain

ii. by a simple edge detector

iii. by the Harris corner detector.

For comparison we select an equal amount of each type of point of interest corresponding to a certain density of the mask, i.e. corresponding to a definite percentage of the pixels where ground truth is available. As edge features we select the locations in f_1 having the largest gradient magnitude $|\nabla f_1|$, where f_1 has been Gaussian smoothed. We thus select those locations where the gradient is larger than the gradient of the quantile determined by the chosen density. In the same way we select the strongest corners in f_1 according to formula (3.1). It has to be noted that the maximal number of corners is lower than the total number of pixels since a corner is defined as a local maximum. As a local neighborhood for determining the maximum we chose a 3×3 image patch centered around each pixel. For both edges and corners the image f_1 is smoothed with a standard deviation of 1. The integration scale ρ for corner detection is taken to be 1 and the parameter k is set to 0.04. These three parameters influence both the performance of the Harris detector and the number of corners found. In literature one can find values for ρ ranging from 0.7 to 3. The higher the value, the less corners are typically found. For our purposes these settings are less stringent and they have not been tuned to obtain the most stable corners. In Figure 6.6 we can see the locations of the 3 types of interest points for the Yosemite sequence with clouds. It shows the 8th frame overlaid with a mask corresponding to two different feature densities.

For the optic flow estimation we use the Horn and Schunck method as in equation (2.7). We include the ground truth information by setting $\mathbf{w}$ to the ground truth in each of the selected points. On the rest of the domain we solve the Euler-Lagrange equations resulting from the energy functional. The total number of SOR-iterations

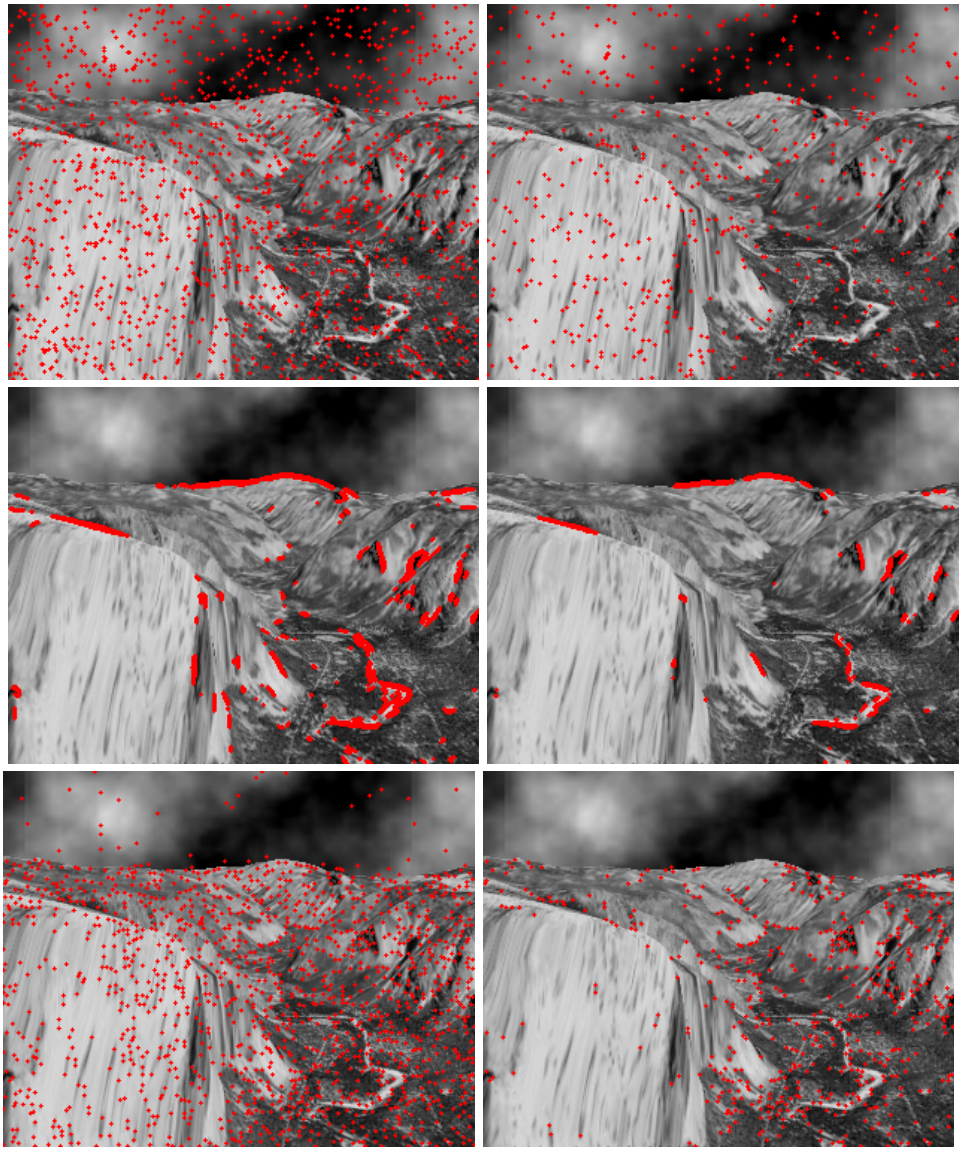

Figure 6.6: *The 3 types of feature points extracted from the 8th frame of the Yosemite sequence with clouds. The left images correspond to a feature density of 1.8% and the right images to a density of 0.7%.* **Top left:** *1426 random points.* **Top right:** *558 random points.* **Middle Left:** *1423 edge points.* **Middle Right:** *554 edge points.* **Bottom left:** *1424 Harris corners.* **Bottom right:** *554 Harris corners.*

is fixed at 500 and the relaxation parameter ω is set to 1.97. The smoothing parameter α and the standard deviation σ from the Horn and Schunck method are optimized with respect to the AAE.

Table 6.1: *Results of a comparative test with 3 types of locations of interest for the Yosemite sequence without clouds. The parameters σ and α are optimized w.r.t. the AAE. 1513 Harris corners were detected corresponding to a density of 2.6%.*

dens. (%)	random			edge			corner		
	σ	α	AAE [°]	σ	α	AAE [°]	σ	α	AAE [°]
0	1.77	805	2.65	1.77	805	2.65	1.77	805	2.65
0.1	1.77	1286	2.38	1.77	835	2.62	1.77	835	2.59
0.2	1.77	2053	2.13	1.77	835	2.61	1.77	898	2.50
0.5	1.77	$> 10^4$	1.43	1.77	898	2.53	1.77	1197	2.24
1	2.36	$> 10^4$	0.98	1.77	964	2.42	1.61	$> 10^4$	1.44
2	3.14	$> 10^4$	0.66	1.77	1240	2.08	1.61	$> 10^4$	0.63
10	3.14	$> 10^4$	0.26	1.77	$> 10^4$	0.78			

In Table 6.1 we collected the results from our tests performed on the Yosemite sequence without clouds. For increasing density of the feature points we list the evolution of the AAE. The density is expressed as a percentage of the pixels where ground truth is available, in this case the 58911 pixels that make up the mountain region. We see clearly that the random addition of ground truth information results in a substantial improvement of the AAE. For a feature density of 1%, roughly corresponding to about 600 features, the AAE has dropped by more than a half. The results are less good for the edge detector. The reason is that we applied a simple threshold on the gradient value resulting in locations with bad spreading. A more reasonable edge detector would produce edges of width one with a better spatial distribution. Although edge features differ from the other two in nature, the test results nevertheless show the importance of a good spatial spread of the locations of interest. This effect can also be observed for the corners. The AAE of the optic flow with Harris corners drops slower than the AAE for random features because the strongest corners will be concentrated in highly textured regions. For higher densities we see that the results are similar to those obtained with randomly distributed features since weaker corners in smoother regions are being found. We also notice the increasing parameter α which indicates the reduced significance of the data term for larger numbers of preset flow vectors. The rate of increase in α depends on the spreading of the features with the fastest growth for random points and the slowest for edge points.

We use the same test setup for the Yosemite sequence with clouds. This sequence brings forth the difficulty of a moving entity without texture and thus poses a challenge for detectors to find stable features. Due to varying illumination and weak gradient values, the clouds also pose difficulties for our data term. Our results are listed in Table 6.2. As expected for edges and corners, the AAE does not drop significantly until locations in the cloud region are being selected. For corners this happens from a density of about 1.7% on. Our simple edge detector only detects high gradient locations in the cloud region for low quantile thresholds and point densities of more than 50%. In Figure 6.7 the resulting flow estimations are shown in color code for a feature density of 1.8%. We can clearly see the singularities in

locations where we include point information. The estimation of the motion of the clouds has a large influence on the overall AAE. If features are detected in the sky region, the optimal smoothness weight is in general very large to compensate for the bad performance of the data term.

Table 6.2: *Results of a comparative test with 3 types of locations of interest for the Yosemite sequence with clouds. The parameters σ and α are optimized w.r.t. the AAE. 1917 Harris corners were detected corresponding to a density of 2.4%.*

dens.	random			edge			corner		
(%)	σ	α	AAE [°]	σ	α	AAE [°]	σ	α	AAE [°]
0	1.33	453	7.17	1.33	453	7.17	1.33	453	7.17
0.1	1.21	1240	6.64	1.21	470	7.19	1.33	486	7.11
0.2	0.91	4870	5.70	1.21	437	7.18	1.33	437	7.08
0.5	1.10	$> 10^4$	4.09	1.21	470	7.16	1.33	562	6.90
1	1.33	$> 10^4$	2.81	1.21	562	7.02	1.33	777	6.47
2	1.61	$> 10^4$	1.92	1.21	750	6.56	1.33	$> 10^4$	2.14
10	1.61	$> 10^4$	< 1.0	1.21	6978	3.88			

Table 6.3: *Results of a comparative test with 3 types of locations of interest for the new Marble sequence. The parameters σ and α are optimized w.r.t. the AAE. 5666 Harris corners were detected corresponding to a density of 2.9%.*

dens.	random			edge			corner		
(%)	σ	α	AAE [°]	σ	α	AAE [°]	σ	α	AAE [°]
0	0.564	835	2.49	0.564	835	2.49	0.564	835	2.49
0.1	0.564	1074	2.43	0.564	964	2.40	0.564	964	2.43
0.2	0.564	1240	2.39	0.564	1074	2.32	0.564	1036	2.38
0.5	0.564	4067	2.054	0.564	1433	2.18	0.564	1486	2.26
1	> 5	$> 10^4$	1.60	0.564	2458	2.00	0.683	3651	2.00
2	> 5	$> 10^4$	1.04	0.683	4532	1.74	> 5	$> 10^4$	0.990
10	> 5	$> 10^4$	< 0.4	0.826	$> 10^4$	0.768			

In the last experiment we use the new Marble sequence. The motion field is everywhere 0 except in two distinct regions with sharp boundaries. The homogeneous regularizer does not perform very well at these sharp boundaries and makes them look a bit blurry. The motion edges coincide with the contours of the moving marble blocks and this may favor the extraction of edge information. From Table 6.3 we can indeed derive that the improvement in AAE is more significant for edge features for low densities. Another reason is that most of the strongest edge points lie on the object boundaries and not within the background region. The background is static and the Horn and Schunck method gives a reasonable estimation there. Therefore the addition of ground truth information outside the two moving blocks, as in the case of random points and corners, is less likely to improve the estimation. For higher feature densities we observe that the random points and the corners result in a lower AAE due to the fact that the added information is better spread. For densities below 1% the corner features are badly spread within the regions of non-vanishing flow because no strong corners are being detected in the shadowed

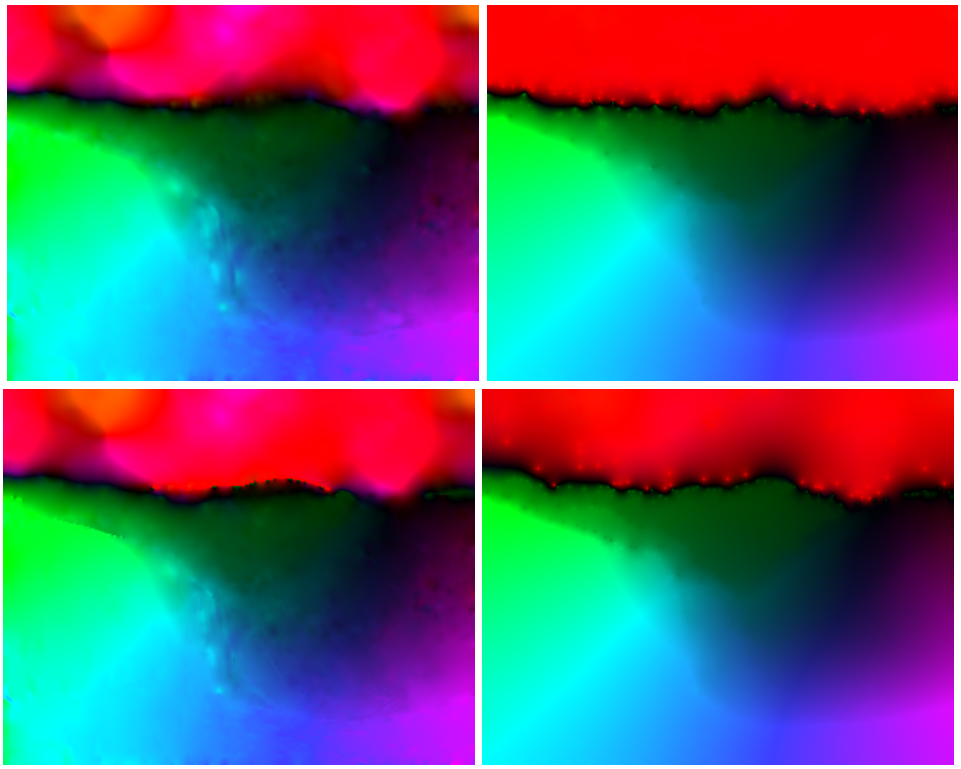

Figure 6.7: *The estimated optic flow field for the Yosemite sequence with clouds. The standard Horn and Schunck method is compared to the Horn and Schunck method with ground truth information added in various types of feature locations with a density of 1.8%* . **Top left:** *The standard Horn and Schunk method.* **Top right:** *With random features.* **Bottom left:** *With edge features.* **Bottom right:** *With Harris corners.*

sides of the marble blocks. This bad spreading is reflected in a poor improvement of the AAE. The influence of ground truth information in edges and the bad spread of Harris corners for low densities can be seen in Figure 6.8. The density of the depicted features is 0.7% . The flow field is a detail of 320×240 pixels enclosing the two moving blocks. The motion boundaries for the corners are determined less well compared to the estimation with ground truth added in the edges.

From the quantitative results of these experiments we conclude that a variational optic flow estimation can benefit from sparse flow information. Our results present an upper limit of how much the Horn and Schunck method can improve from predetermined flow information. In practice the preset displacements are produced by an actual feature detector and therefore deviations from ground truth are to be expected. Qualitatively we can derive obvious conditions for the locations in which we wish to know the displacement beforehand. These locations have to be dense enough and well distributed over the area of motion. The cloud region in the Yosemite sequence proves that lack of texture can pose a problem for simple gradient based features. But even if the feature density was lower there, we observed a significant improvement in the AAE if the smoothness weight was chosen high enough. This leads us to the conclusion that the density of features does not have to be very high within a certain region as long as the motion is uniform.

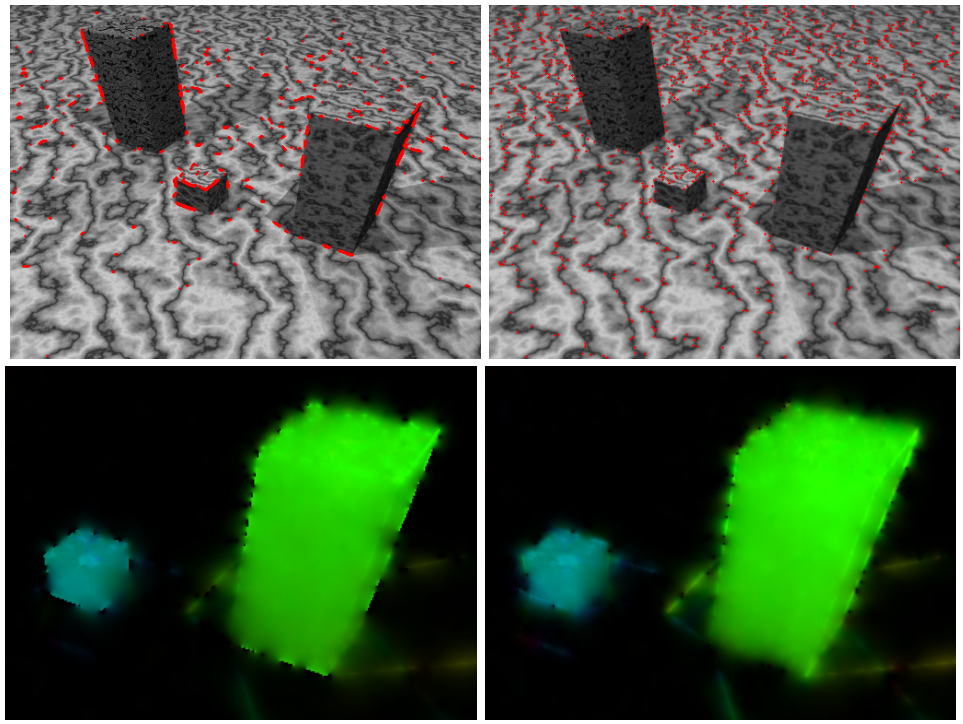

Figure 6.8: *The edge and corner locations detected in frame 150 of the new Marble sequence and the corresponding optic flow field estimated by the Horn and Schunck method with ground truth information added in the same feature locations.* **Top left:** *1373 edge points.* **Top right:** *1373 Harris corners.* **Bottom left:** *The estimated optic flow with ground truth added in the edge locations.* **Bottom right:** *The estimated optic flow with ground truth added in the corner locations.*

6.5 Experiments with SIFT features

SIFT features have a spatial distribution that is similar to that of Harris corners. In this section we will use them to establish a sparse motion field from which we can estimate a dense optic flow.

6.5.1 An evaluation of SIFT for Motion Estimation

First we summarize some basic statistics of the SIFT algorithm when applied to motion estimation for the four test sequences. After that we will asses the dense flow estimation obtained by directly inserting the sparse motion field established by SIFT into the Horn and Schunck method.

Basic Assessment of SIFT

In Table 6.4 we list the total number of features that are detected by the SIFT algorithm for the four frame pairs that we consider. Additionally, we list the number of correspondences that are found between the frames of each pair by comparing the Euclidean distance of the feature descriptors. The thresholds that we used as distance ratio between the closest and the second closest point in the descriptor space are 0.6 and 0.8.

Table 6.4: *The upper part of the table shows the number of features per frame that are found by the SIFT algorithm. The lower part of the table lists the total number of SIFT features that have been matched for a matching threshold of 0.6 and 0.8. The densities of the effective matches are listed w.r.t. the total number of pixels where ground truth is available.*

image sequence	Yosemite without clouds		Yosemite with clouds		old Marble		new Marble	
frame number	8	9	8	9	16	17	150	151
# SIFT features	802	786	821	804	5492	5432	4032	4084
threshold	0.6	0.8	0.6	0.8	0.6	0.8	0.6	0.8
# matches	556	588	581	612	4191	4441	3518	3596
# eff. matches	488	515	513	539	3376	3560	2830	2893
mask density (%)	0.83	0.87	0.64	0.68	1.5	1.6	1.4	1.5

As mentioned in Chapter 3 multiple SIFT features can be found in the same spatial location. As a consequence a feature location can be matched to multiple other locations and multiple matches between the same feature locations can occur. We have experienced that the latter case occurs much more frequently and these multiple matches result in the same displacement vector in a certain location. If features with the same position happen to be matched to features with a different position we use the correspondence that has been established last and has not been identified as an outlier. By following this approach we assume that the variance in a SIFT displacement $\mathbf{d}_i$ caused by choosing from multiple matches is inherent to the feature detector and similar in nature to the variance caused by localization errors.

Eventually we end up with one displacement per matched feature location. Therefore the number of effective matches is less than than the total number of correspondences found by comparing the feature descriptors. The number of effective matches together with the according mask density is listed in Table 6.4. The effective matches are used to determine the prescribed sparse displacement field. We see that a large fraction of the established feature correspondences are assigned to identical spatial locations. For the larger image sizes the number of effective matches can be up to 20% less than the total number of found matches.

In Figure 6.9 the sparse displacement field that results from the set of effective matches is superimposed on the first frame of each image pair. This is done for a distance ratio threshold of 0.8. We can observe spurious displacements for all the image pairs. For a threshold of 0.6 no severe outliers can be visually detected for the Yosemite sequences while for the other sequences one or two large mismatches are still clearly present. Severe outliers can easily be removed by disregarding displacements with an excessive magnitude.

SIFT versus Horn and Schunck

Table 6.5 lists the quantitative differences between the sets of SIFT displacements that result from applying different distance thresholds. The AAE and the ASE of the set of SIFT displacements have been calculated and compared to the AAE and the ASE of a sparse optic flow estimation. This means that they are compared to the AAE and the ASE of a set of displacements in the same locations, but

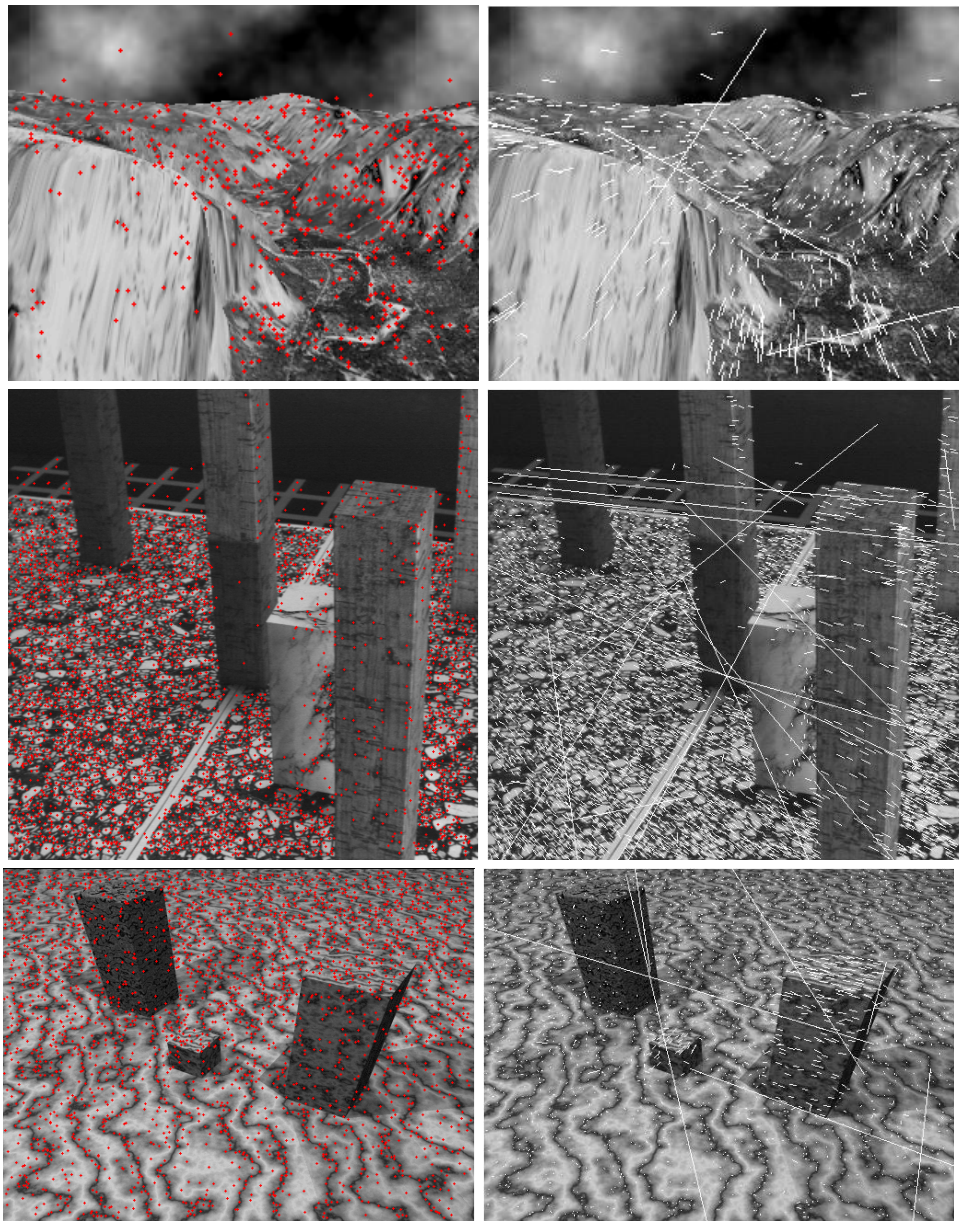

Figure 6.9: *The SIFT feature locations and displacements for a distance ratio threshold of 0.8. The displacement vectors are depicted at 4 times their original size.* **Top left:** *The locations for frame 8 of the Yosemite sequence with clouds.* **Top right:** *The displacements for the Yosemite sequence with clouds.* **Middle Left:** *The locations for frame 16 of the old Marble sequence.* **Middle Right:** *The displacements for the old Marble sequence.* **Bottom left:** *The locations for frame 150 of the new Marble sequence.* **Bottom right:** *The displacements for the new Marble sequence.*

Table 6.5: *Evaluation of the average angular error AAE_{SIFT} and the average squared error ASE_{SIFT} of the SIFT displacements for a distance ratio threshold of 0.6 and 0.8. They are compared with the AAE and the ASE of a standard Horn and Schunck estimation in the locations of the SIFT features.*

image sequence	Yosemite without clouds		Yosemite with clouds		old Marble		new Marble	
threshold	0.6	0.8	0.6	0.8	0.6	0.8	0.6	0.8
AAE_{SIFT} [°]	7.35	8.23	7.63	8.46	6.49	6.867	4.46	4.84
AAE_{HS} [°]	2.25	2.28	4.40	4.54	5.14	5.215	1.63	1.81
ASE_{SIFT}	0.091	0.37	0.10	0.33	34	40	0.037	0.059
ASE_{HS}	0.012	0.013	0.055	0.057	33	40	0.010	0.014

resulting from a standard optic flow estimation. The optic flow is obtained with the method of Horn and Schunck with optimized parameters. Large outliers were removed from each set of SIFT displacements by applying a threshold of 10 pixels to the magnitude. The old Marble sequence lost 1 and 15 effective matches for the respective thresholds of 0.6 and 0.8, while the other sequences only lost a few effective matches for a threshold of 0.8.

The AAE of the SIFT displacements is in all cases larger than the AAE of the estimated optic flow in the corresponding locations. Also the ASE follows this trend. We might get discouraged by these results to use SIFT in a combination with a standard optic flow method because it does not have the appearance of improving the estimation. We further notice that the ASE differs strongly between the two ratio thresholds for the Yosemite sequences. This indicates that we still have some outlying data for the 0.8 threshold that deviates more from ground truth in the L_2-norm than the bulk of the data.

We can get an additional impression of the localization noise that comes with the SIFT features by taking a look at the background of the new Marble sequence. We know that this background is static but a large percentage of the SIFT features are not being detected in exactly the same location in both frames. Hence many SIFT displacements in the background region deviate slightly from zero, giving rise to measurement noise.

Combining SIFT with Horn and Schunck

In Table 6.6 we give the results of an experiment conducted with ground truth as in the previous section. We set $\mathbf{w}$ to the ground truth in each of the SIFT feature locations and on the rest of the domain we solve the Euler-Lagrange equations of the Horn and Schunck method. The total number of SOR-iterations is fixed at 500 and the relaxation parameter ω is set to 1.97 . The smoothing parameter α and the standard deviation σ are optimized with respect to the AAE. We compare the results with those of the same experiment but with $\mathbf{w}$ set to the actual displacement of the SIFT feature in the corresponding location. SIFT locations with a displacement larger than 10 pixels were not taken into account in both experiments.

The improvements due to the use of ground truth can be substantial but the results for the two distance ratios do not differ much. For the Yosemite sequence with clouds the results are even identical because an increase in threshold to 0.8 does not produce more features in the cloud region. Combining the Horn and Schunck method directly with SIFT produces AAEs that are worse than those

of the standard method. The regression in quality is for most image sequences limited but the results are far from the maximal improvement that is dictated by the ground truth experiment. Opposed to the use of ground truth, the use of SIFT displacements demands a smaller optimal smoothness weight. This indicates that the propagation of information from the sample values is limited.

Table 6.6: *The results for the Horn and Schunck (*HS*) method combined with ground truth data and SIFT. The AAEs are compared to those of the standard method. The parameters σ and α are optimized w.r.t the AAE.*

image sequence	method	threshold	σ	α	AAE [°]
Yosemite without clouds	standard HS	-	1.77	805	2.65
	with ground truth	0.6	1.77	7773	1.80
	with SIFT		1.61	626	3.17
	with ground truth	0.8	1.77	8659	1.74
	with SIFT		1.61	453	3.66
Yosemite with clouds	standard HS	-	1.33	453	7.17
	with ground truth	0.6	1.46	2458	6.13
	with SIFT		1.33	486	7.35
	with ground truth	0.8	1.46	2287	6.13
	with SIFT		1.33	339	7.78
old Marble	standard HS	-	2.59	1000	5.30
	with ground truth	0.6	> 5	4216	1.78
	with SIFT		2.59	865	5.50
	with ground truth	0.8	> 5	$> 10^4$	1.73
	with SIFT		2.36	865	5.56
new Marble	standard HS	-	0.564	835	2.49
	with ground truth	0.6	> 5	$> 10^4$	1.33
	with SIFT		0.564	562	2.88
	with ground truth	0.8	> 5	$> 10^4$	1.27
	with SIFT		0.564	486	2.97

Despite the threshold on the magnitude, there are still some outliers present in the data set for a distance threshold of 0.8. This can be inferred from the higher AAE but it can also be inspected visually. Figure 6.10 displays the optic flow field resulting from our experiments for the Yosemite sequence with clouds. We clearly see the data points emerging as singularities in the flow field. Below each flow field the angular error is displayed with the mask of SIFT features superimposed. We observe that several SIFT displacement vectors introduce local errors in the flow angle. For a distance threshold of 0.6 most deviations from the true flow are fine grained and very local. The flow field for the higher threshold is disturbed by some bigger outliers that require a removal strategy that is more involved than just selecting on magnitude.

In a final series of experiments we try to get an idea of the number of SIFT correspondences that give a displacement which is close to ground truth. At the same time we try to extract a subset of SIFT displacements that is able to improve the AAE of the standard Horn and Schunck method. To this end we compare all SIFT displacement vectors with the ground truth vector in the same location. The vectors that are close to ground truth according to some measure are retained and used in a similar test setup as the previous ones. The selection of the subset

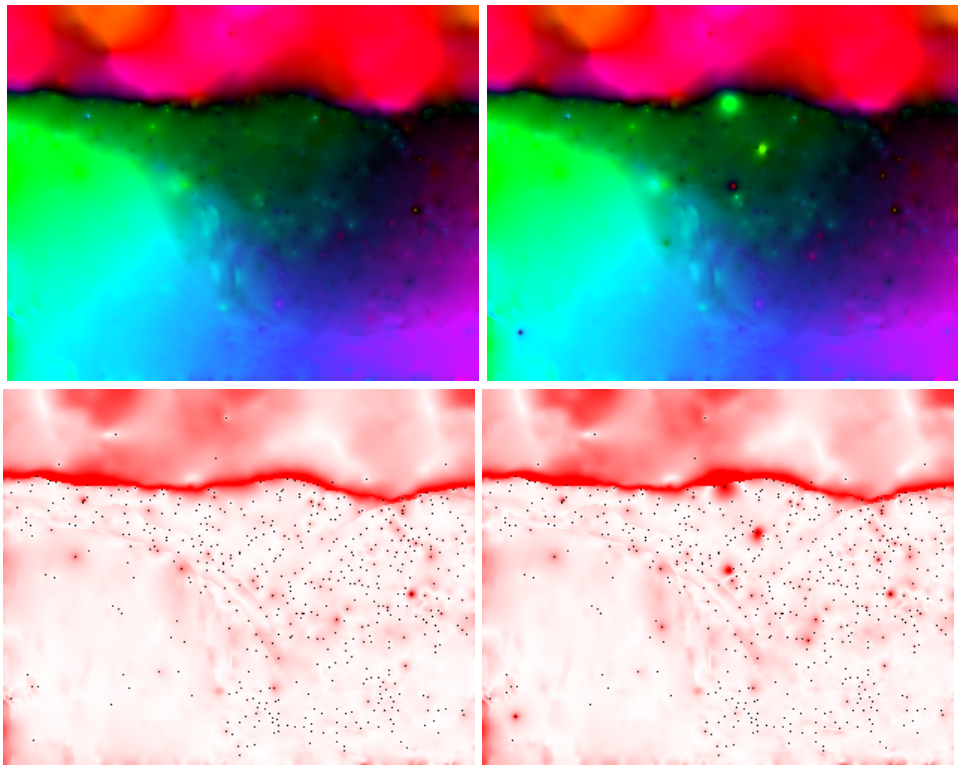

Figure 6.10: *Combining the Horn and Schunck method with a data set of SIFT displacement vectors. The bottom images display the angular error where a brighter color indicates a larger error.* **Top left:** *Flow field for a distance threshold of 0.6.* **Top right:** *Flow field for a distance threshold of 0.8.* **Bottom left:** *Angular error for a distance threshold of 0.6.* **Bottom right:** *Angular error for a distance threshold of 0.8.*

will be based on the improvement of the AAE. If the set is selected too small, the improvement in AAE will be negligible. If the set is chosen too large, the AAE might get worse due to displacements that deviate too much from ground truth.

To determine if a displacement vector is close to ground truth we test if it lies within a certain radius of the ground truth vector. We chose this radius to be 20% of the length of the respective ground truth vector since this resulted in the best improvements in AAE for most test sequences. For the new Marble sequence we selected a subset on the basis of the individual angular errors ($< 3°$) because the major part of the true flow field is 0 and so would be the cut-off radius. Table 6.7 lists the resulting number of effective matches that give a displacement which is close to ground truth for a ratio threshold of 0.6. This subset of displacements is further used in an optic flow estimation by means of the Horn and Schunck method. We set $\mathbf{w}$ in each of the retained SIFT points to the corresponding SIFT displacement and on the rest of the domain we solve the Euler-Lagrange equations. The table shows the AAE of the estimations and these can be compared with the results from Table 6.6. We notice an improvement in AAE for all image sequences but the improvements are not as good as the ones obtained with ground truth (note that there were more effective matches in the ground truth experiment, thus the comparison does not fully hold). The set of SIFT displacements that are responsible for these improvements are between 50 and 60% of the total number of displacements.

Table 6.7: *The results for the Horn and Schunck method combined with a subset of the SIFT displacements. The set of SIFT displacements has been obtained by means of ground truth. The parameters σ and α are optimized w.r.t the AAE of the combined method.*

image sequence	Yosemite without clouds	Yosemite with clouds	old Marble	new Marble
# eff. matches	252	266	2291	1647
σ	1.77	1.33	3.14	0.622
α	1197	835	1596	1113
AAE[°]	2.30	6.73	4.75	2.43

In Table 6.8 we additionally list the AAE and the ASE of the subset of SIFT displacements selected using ground truth. These are compared to the AAE and the ASE of a set of displacements in the same locations, but resulting from the standard Horn and Schunck estimation. The ASE of the SIFT displacements is not worse than the ASE of the Horn and Schunck method. This is not surprising since the selection of the subset was based on the Euclidean distance. An interesting observation from this table is that an improvement in AAE is possible by combining SIFT and standard variational optic flow, despite the AAE of the SIFT displacements being worse than the AAE of the variational method in the same locations. For the Yosemite sequence without clouds $\mathrm{AAE}_{\mathrm{SIFT}} > \mathrm{AAE}_{\mathrm{HS}}$, but the overall AAE of the combined method drops from 2.64 to 2.30. The AAE of our sample of SIFT displacements is obviously not a direct indication of the quality of the optic flow field that results from the data fitting process.

Table 6.8: *The AAE and the ASE of the subset of SIFT displacements that has been selected by means of ground truth. They are compared to the AAE and the ASE of the standard Horn and Schunck estimation in same locations.*

image sequence	Yosemite without clouds	Yosemite with clouds	old Marble	new Marble
$\mathrm{AAE}_{\mathrm{SIFT}}$ [°]	2.95	3.00	4.35	1.32
$\mathrm{AAE}_{\mathrm{HS}}$ [°]	2.21	4.77	4.71	0.82
$\mathrm{ASE}_{\mathrm{SIFT}}$	0.018	0.021	32	0.0012
$\mathrm{ASE}_{\mathrm{HS}}$	0.018	0.076	32	0.0018

6.5.2 Approximation versus Interpolation

Up till now we have been using the exact values of the SIFT displacements. We have used SIFT together with the Horn and Schunck method by setting the optic flow $\mathbf{w}$ to the values of the SIFT displacements in the feature locations and by solving the Euler-Lagrange equations on the rest of the domain. So far, the result was a minor deterioration of the optic flow estimation if we used the complete set of SIFT displacements. There are however two possibilities of improving the quality of our estimation:

- ◇ Evidently, the AAE of the SIFT displacements is increased due to a small amount of bad samples that however have less influence on the overall AAE when we combine SIFT with the Horn and Schunck method. If we provide robustness against these bad samples their influence can be lowered further.
- ◇ We can avoid exact fitting of the data by imposing additional smoothness in the interpolations points, thereby performing an approximation instead of an interpolation. Like this we try to capture a general trend in the sample set without giving every single SIFT displacement a full weight.

Our two new models for optic flow estimation, the feature-based approximation method and the combined optic flow method, avoid exact fitting of the data by introducing smoothness in the data points. Additionally, they provide a robust similarity term that downgrades the importance of values that vary too much from the local estimate. We point out that the robust similarity term will not have any effect if we perform a pure interpolation of the data instead of an approximation, because in the case of interpolation the values in the data points are not allowed to vary from the measurements.

Robust Approximation versus Interpolation

In the following we will asses the performance of the two paradigms presented above. Therefore we compare our feature-based approximation method from Chapter 3 with the outcome of pure interpolation of the SIFT displacements. Interpolation is equal to approximation with a smoothness weight of 0 in the data points and is thus a special case of our feature-based approximation scheme. We also compare our combined method of Chapter 4 with the results that were obtained in Table 6.6 by setting $\mathbf{w}$ to the exact SIFT displacements and solving the Euler-Lagrange equations on the rest of the domain. The latter approach can be considered as a special case of our combined optic flow method since the exact SIFT displacements are interpolated with a standard optic flow method. To adopt a consistent nomenclature we will call the general combined optic flow method, which has a robust similarity term and smoothness in the data points, the robust combined method.

In our experiments we use a set of SIFT displacements that is obtained by applying a distance ratio threshold of 0.6. This threshold provides for most image sequences a maximal number of point correspondences without contamination by cumbersome mismatches. SIFT displacements with a magnitude larger than 10 pixels have been removed. We use a robust similarity term with the L_1-penalization as in (3.27)-(3.28) and the homogeneous operator.

In Tables 6.9 - 6.12 we summarize the results of our comparative tests for the Yosemite sequences and the Marble sequences. The tables list the results of an optic flow estimation with the standard method, with interpolation of SIFT displacements, with interpolation and post-smoothing, with the feature-based approximation scheme (3.12)-(3.13), with interpolation of the SIFT displacements by standard optic flow and with the robust combined optic flow method (4.7)-(4.8). For the homogeneous operator the standard optic flow estimation amounts to the classical Horn and Schunck method. The total number of SOR-iterations was fixed at 500 and the relaxation parameter ω was set to 1.97 .

Table 6.9: *Results for the Yosemite sequence without clouds. The estimation methods are the standard optic flow method, the interpolation and robust approximation of SIFT displacements, the interpolation of SIFT displacements with standard optic flow and the robust combined optic flow method. The parameters c, σ and the smoothness weight are optimized w.r.t. the AAE.*

method	c	σ	smooth. weight	AAE [°]
optic flow	-	1.77	805	2.65
interpolation	-	-	-	3.82
interpolation + post-smoothing	0.025	-	-	3.49
robust approximation	-	-	1.43	3.75
SIFT + optic flow	-	1.61	626	3.17
robust combined optic flow	0.98	1.77	800	2.60

Table 6.10: *Results for the Yosemite sequence with clouds. The estimation methods are the standard optic flow method, the interpolation and robust approximation of SIFT displacements, the interpolation of SIFT displacements with standard optic flow and the robust combined optic flow method. The parameters c, σ and the smoothness weight are optimized w.r.t. the AAE.*

method	c	σ	smooth. weight	AAE [°]
optic flow	-	1.33	453	7.17
interpolation	-	-	-	14.4
interpolation + post-smoothing	0.025	-	-	14.4
robust approximation	-	-	0	14.4
SIFT + optic flow	-	1.33	486	7.35
robust combined optic flow	0.99	1.33	550	7.14

Table 6.11: *Results for the old Marble sequence. The estimation methods are the standard optic flow method, the interpolation and robust approximation of SIFT displacements, the interpolation of SIFT displacements with standard optic flow and the robust combined optic flow method. The parameters c, σ and the smoothness weight are optimized w.r.t. the AAE.*

method	c	σ	smooth. weight	AAE [°]
optic flow	-	2.59	1000	5.30
interpolation	-	-	-	5.76
interpolation + post-smoothing	0.010	-	-	5.48
robust approximation	-	-	0	5.76
SIFT + optic flow	-	2.59	865	5.50
robust combined optic flow	0.999	2.59	1040	5.19

Table 6.12: *Results for the new Marble sequence. The estimation methods are the standard optic flow method, the interpolation and robust approximation of SIFT displacements, the interpolation of SIFT displacements with standard optic flow and the robust combined optic flow method. The parameters c, σ and the smoothness weight are optimized w.r.t. the AAE.*

method	c	σ	smooth. weight	AAE [°]
optic flow	-	0.564	835	2.49
interpolation	-	-	-	3.57
interpolation + post-smoothing	0.035	-	-	3.43
robust approximation	-	-	3.40	3.29
SIFT + optic flow	-	0.564	562	2.88
robust combined optic flow	0	0.564	835	2.49

We observe from the tables that pure interpolation of the data values performs worse than the Horn and Schunck method. Subsequent smoothing of the flow field reduces the AAE in general by a small amount. The weight c in the tables refers to the same c as in the equations for post-smoothing (3.25)-(3.26).The final result of smoothing the interpolant comes close to the result of the standard optic flow estimation for the old Marble sequence. This outcome can be considered as surprisingly good. For this sequence and for the Yosemite sequence with clouds, the approximation scheme performs worse than pure interpolation because of some isolated points. These isolated points can be crucial for an accurate optic flow estimation, but their contribution decreases a lot with increasing smoothness weight. Robust approximation gave the best results of all feature-based methods for the new Marble sequence with a final AAE of 3.29°. In Figure 6.11 we can see how interpolation with homogeneous diffusion results in singularities while the approximation scheme tends to give smoother results. Interpolation and approximation of SIFT displacement data has an advantage at frame boundaries where standard optic flow methods typically suffer from outliers due to occlusions.

The robust combined optic flow model yields a small improvement of the Horn and Schunck method for all image sequences, except for the new Marble sequence. The drop in AAE is very small but it confirms our earlier conclusion that the error in the data set does not reflect the error of the resulting flow estimation. For the combined method the weight c in the tables refers to the value of $c(\mathbf{x})$ in the SIFT feature locations. Visually there is no distinction between the flow field that is estimated by the combined method and the flow field that is estimated by the standard method. The smoothness weight is relatively large in the feature locations and this prevents the SIFT displacements from appearing as singularities. The proportion of data term and smoothness term is more or less equal to the one for the standard optic flow method. With our robust combined optic flow method we were able to improve the results of Table 6.6 considerably for most image sequences.

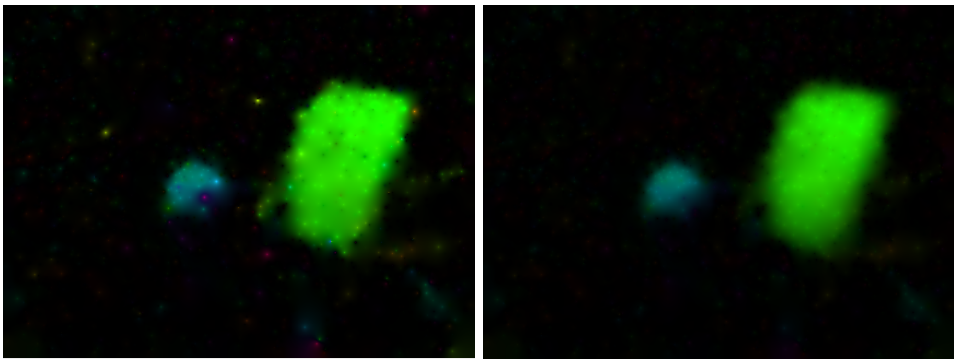

Figure 6.11: *Interpolation of SIFT displacements with the homogeneous operator for the new Marble sequence.* **Left:** *Pure interpolation, AAE = 3.57°.* **Right:** *Approximation, AAE = 3.29°.*

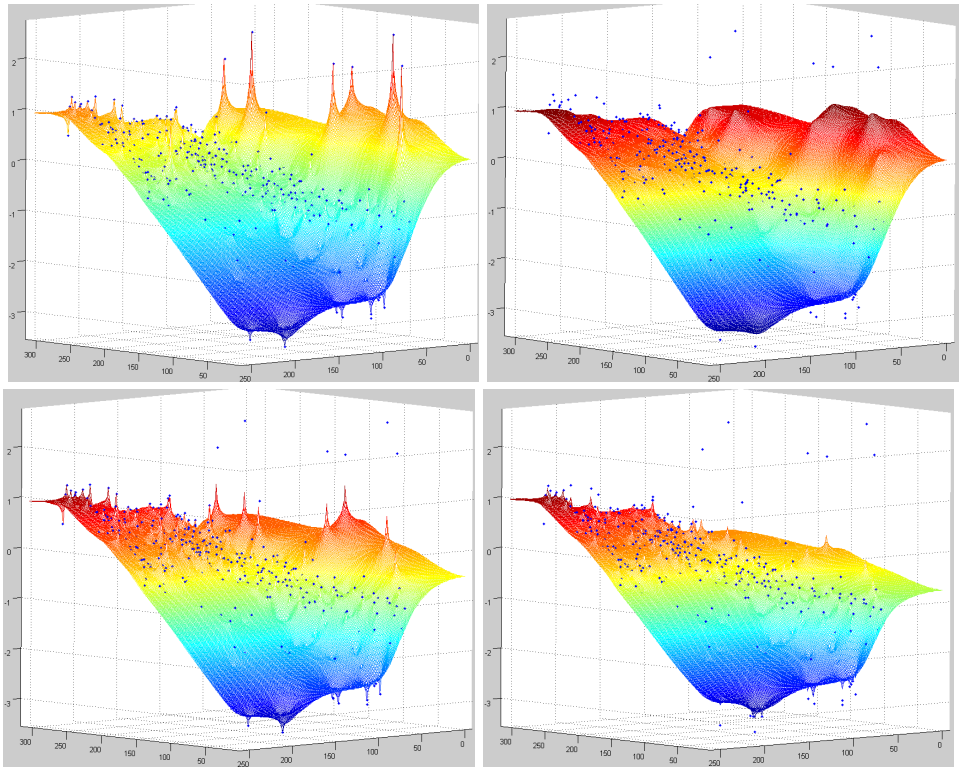

Figure 6.12: *The u-component of the flow estimation with the homogeneous operator for the Yosemite sequence with clouds. The u-components of the SIFT displacements are displayed as blue diamonds. The high point density region in the foreground/left is the mountain part of the sequence, while the isolated high values in the back indicate the sky region.* **Top left:** *Pure interpolation.* **Top right:** *Smoothed interpolant.* **Bottom left:** *Approximation with a smoothness weight of 0.8.* **Bottom right:** *Approximation with a smoothness weight of 2.*

Robust Approximation versus Interpolation with Post-smoothing

Interpolation with post-smoothing and robust approximation of the SIFT displacements provide similar results if the data set is not too sparse. However, the performance of both approaches differs drastically in regions with a low data density like the cloud region of the Yosemite sequence with clouds. Figure 6.12 displays the u-component of the estimated optic flow field for this sequence. We see how interpolation by homogeneous diffusion creates large singularities in the cloud region. Robust approximation with increasing smoothness weight does not give better results because it smooths out the singularities in the sky region, which eventually results in the absence of motion. The optic flow looks different if we post-smooth the interpolant as explained in 3.4.4. Despite a gross underestimation of the cloud motion, post-smoothing preforms far better than approximation.

The feature-based approximation scheme of 3.4.3 can also suffer from a numerical drawback. Data propagation between subsequent SOR steps is slow if the smoothness weight is chosen to be large. This is because data generation only takes place in very few pixels, namely there where the similarity term exists. A very smooth end result will usually require a large number of iterations unless the flow is initialized closely to the solution.

We have justified post-smoothing of the interpolant as a means of reducing noise that is present in the sample set of SIFT-displacements. The comparison with other methods is not completely fair since we can smooth their estimates too. The outcome of a Horn and Schunck method will in general also profit from a post-smoothing step. We will, however, reserve this post-processing step solely for scattered data interpolation because its rationale grew from the idea to develop an approximation method that does not fit the data exactly.

6.5.3 An Evaluation of Different Operators

We have presented results for our robust approximation scheme and our robust combined method for the homogeneous operator. In the following we will show how other choices of the elliptic operator can affect the outcome. We used the same test setup as in the previous experiments.

Biharmonic and Mixed Operator

The motivation to use biharmonic smoothing comes from the fact that it does not create singularities in the interpolation points. On the other hand, results seldom improve unless the biharmonic operator is combined with another operator like the homogeneous one. We can see from Table 6.14 that a standard optic flow estimation with a biharmonic smoothness term gave a worse AAE than the Horn and Schunck method. The same is true for biharmonic interpolation of the SIFT displacements. The main reason for the deterioration in AAE is that over- and undershoots occur which is typical for higher order spline interpolation. Figure 6.13 shows the biharmonic interpolation of the SIFT displacements for the Yosemite sequence with clouds. The over- and undershoots are clearly visible in the resulting flow field and in the plot of the u-component.

Tables 6.13 - 6.15 list the results for the mixed operator that results from combining homogeneous and biharmonic smoothing:

$$\alpha \cdot \Delta + \beta \cdot \Delta^2 \quad .$$

Table 6.13: *Results for the Yosemite sequence without clouds. The estimation methods are the standard optic flow method, the interpolation of SIFT features with post-smoothing, the robust approximation method and the robust combined optic flow method. The smoothing operator is the homogeneous or the combined homogeneous-biharmonic operator. The parameters c, σ and the smoothness weights are optimized w.r.t. the AAE.*

method	smooth. operator	c	σ	hom. weight	biharm. weight	AAE [°]
optic flow	hom.	-	1.77	805	-	2.65
	mixed	-	1.77	805	-	**2.65**
interpolation	hom.	0.025	-	-	-	3.49
+ post-smoothing	*mixed.*	0.025	-	1	0	**3.49**
robust	hom.	-	-	1.43	-	3.75
approximation	*mixed*	-	-	1.33	0.138	**3.72**
robust combined	hom.	0.98	1.77	800	-	2.60
optic flow	*mixed*	0.98	1.77	800	2500	**2.54**

Table 6.14: *Results for the Yosemite sequence with clouds. The estimation methods are the standard optic flow method, the interpolation of SIFT features with post-smoothing, the robust approximation method and the robust combined optic flow method. The smoothing operator is the homogeneous, the biharmonic or the combined homogeneous-biharmonic operator. The parameters c, σ and the smoothness weights are optimized w.r.t. the AAE.*

method	smooth. operator	c	σ	hom. weight	biharm. weight	AAE [°]
optic flow	hom.	-	1.33	453	-	7.17
	biharm.	-	1.33	-	560	**9.15**
	mixed.	-	1.33	453	900	**7.12**
interpolation	hom.	0.025	-	-	-	14.4
+ post-smoothing	*mixed.*	0.025	-	1	0.75	**12.3**
robust	hom.	-	-	0	-	14.4
approximation	*mixed*	-	-	0	0	**12.3**
robust combined	hom.	0.990	1.33	550	-	7.14
optic flow	*mixed*	0.996	1.33	535	1057	**7.10**

This operator has two smoothness weights and the values of both weights are listed in the tables. For pure interpolation only their ratio outside the data points matters. The solution of the biharmonic operator converges very slowly and in order to obtain a stable solution we had to set the number of iterations to 2000.

For the Yosemite sequence without clouds the approximation scheme seems to benefit slightly from the mixed operator while the interpolation results improve for the Yosemite sequence with clouds. In Figure 6.13 we see that the sky region gets a bit more filled by the mixed interpolant but the overall filling-in is still insubstantial. For both Marble sequences the feature-based methods profit from the mixed operator .

Table 6.15: *Results for the old Marble sequence. The estimation methods are the standard optic flow method, the interpolation of SIFT features with post-smoothing, the robust approximation method and the robust combined optic flow method. The smoothing operator is the homogeneous or the combined homogeneous-biharmonic operator. The parameters c, σ and the smoothness weights are optimized w.r.t. the AAE.*

method	smooth. operator	c	σ	hom. weight	biharm. weight	AAE [°]
optic flow	hom.	-	2.59	1000	-	5.30
	mixed	-	2.59	1074	805	**5.17**
interpolation + post-smoothing	hom.	0.01	-	-	-	5.48
	mixed.	0.01	-	1	0	**5.48**
robust approximation	hom.	-	-	0	-	5.76
	mixed	-	-	0	0	**5.76**
robust combined optic flow	hom.	0.999	2.59	1040	-	5.19
	mixed	0.990	2.59	1040	2916	**5.05**

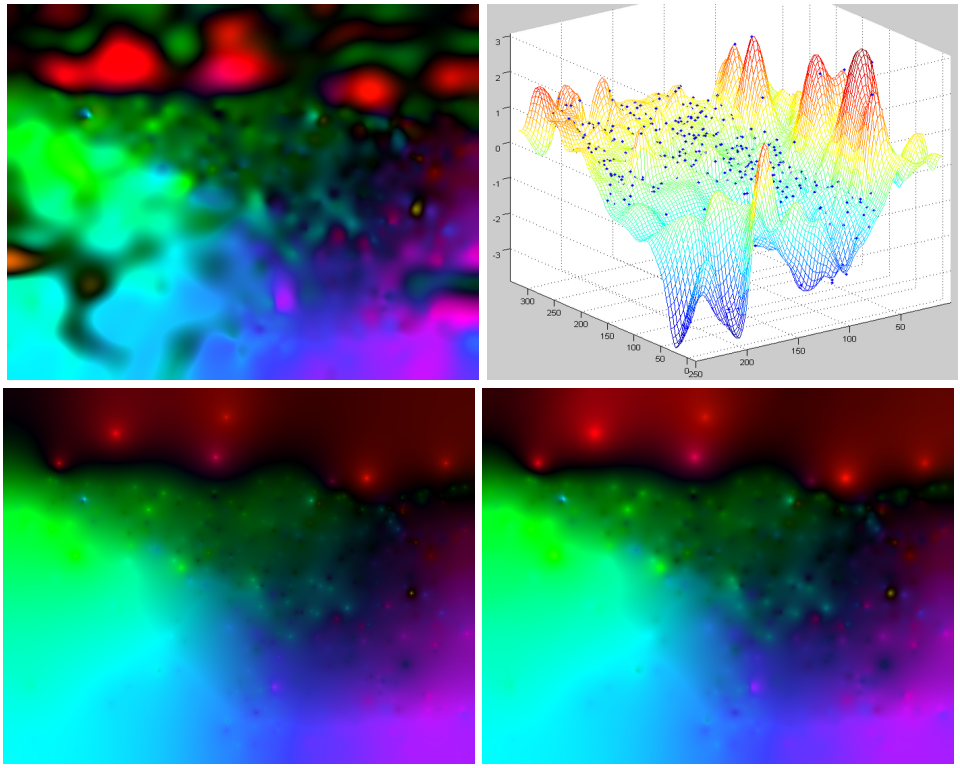

Figure 6.13: *Biharmonic and mixed interpolation of the SIFT displacements for the Yosemite sequence with clouds* **Top left:** *Biharmonic interpolation, AAE* $> 23°$. **Top right:** *Biharmonic interpolation, u-component.* **Bottom left:** *Homogeneous interpolation, AAE = 14.4°.* **Bottom right:** *Mixed homogeneous-biharmonic interpolation, AAE = 12.6°.*

The standard optic flow estimation with the mixed regularizer tends to be slightly better than the Horn and Schunck method for most image sequences. The robust combined optic flow method provides a modest improvement of AAE over the standard method.

Flow-driven Operator

Adding the biharmonic operator did not give significant improvements over the results from the homgeneous operator. Therefore we take a look at EED which has proven effectiveness in image interpolation and inpainting. As robust approximation scheme we use (3.19)-(3.20) with the Charbonnier diffusivity (3.18). The equations of the corresponding standard optic flow method do not minimize an energy functional. Despite being PDE-based it cannot be termed a variational method.

Table 6.16: *Results for the new Marble sequence. The estimation methods are the standard optic flow method, the interpolation of SIFT features with post-smoothing, the robust approximation method and the robust combined optic flow method. The smoothing operator is the homogeneous operator or EED. The parameters c, σ, λ and the smoothness weight are optimized w.r.t. the AAE.*

method	smooth. operator	c	σ	λ	smooth. weight	AAE [°]
optic flow	hom.	-	0.564	-	835	2.49
	EED	-	0.564	0.004	$> 10^4$	**1.26**
interpolation	hom.	-	-	-	-	3.57
interpol.+post-smooth.		0.035	-	-	-	3.43
interpolation	*EED*	-	-	0.009	-	**2.79**
interpol.+post-smooth.		0.003	-	0.009	-	**1.76**
robust approximation	hom.	-	-	-	3.40	3.29
	EED	-	-	0.009	18.4	**1.59**
robust combined optic flow	hom.	0	0.564	-	835	2.49
	EED	0.996	0.564	0.004	$> 10^4$	**1.25**

Dense motion estimation from SIFT displacements profits the most from EED for the new Marble sequence. In Table 6.16 we see that the robust approximation scheme with EED estimates an optic flow field with an AAE of only 1.59°. This is about 40% better than the classical Horn and Schunck estimation. The standard optic flow method with EED is almost 50% better than the Horn and Schunck method, thus outperforming our approximation scheme. This indicates that a similarity constraint with SIFT cannot substitute the simple brightness constancy assumption. The robust combined method was hardly able to improve upon the standard optic flow for this sequence.

Figure 6.14 shows the estimated flow fields for the new Marble sequence. The optic flow that is estimated by robust approximation is visually less attractive than the standard estimation. Boundaries in the flow field seem to correspond to the convex hull of the non-zero SIFT displacements instead of the actual motion boundaries. In the figure we added the optic flow obtained by interpolating the ground truth values in the SIFT features with EED. This gave an AAE of 0.46°. EED interpolation of the ground truth establishes a lower limit on the accuracy of the feature-based method for the new Marble sequence. It additionally illustrates the interpolating capability of EED.

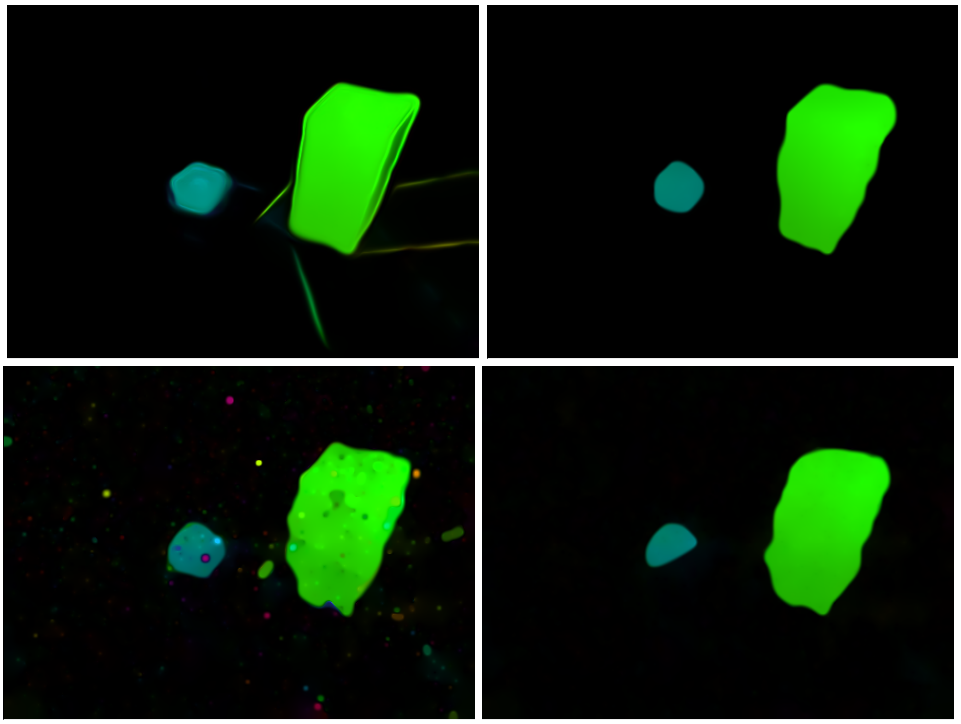

Figure 6.14: *Motion estimation using EED for the new Marble sequence.* **Top left:** *Standard optic flow, AAE = 1.26°.* **Top right:** *Interpolation of the ground truth in the SIFT feature locations, AAE = 0.46°.* **Bottom left:** *Interpolation of the SIFT displacements, AAE = 2.79°.* **Bottom right:** *Approximation with SIFT displacements, AAE = 1.59°.*

Table 6.17: *Results for the old Marble sequence. The estimation methods are the standard optic flow method, the interpolation of SIFT features with post-smoothing, the robust approximation method and the robust combined optic flow method. The smoothing operator is the homogeneous operator or EED. The parameters c, σ, λ and the smoothness weight are optimized w.r.t. the AAE.*

method	smooth. operator	c	σ	λ	smooth. weight	AAE [°]
optic flow	hom.	-	2.59	-	1000	5.30
	EED	-	2.59	0.02	1654	**5.02**
interpolation	hom.	-	-	-	-	5.76
interpol. + post-smooth.		0.010	-	-	-	5.48
interpolation	*EED*	-	-	>1	-	**5.76**
interpol. + post-smooth.		0.010	-	>1	-	**5.48**
robust approximation	hom.	-	-	-	0	5.76
	EED	-	-	0.02	5.4	**5.67**
robust combined optic flow	hom.	0.999	2.59	-	1040	5.19
	EED	0.990	2.59	0.02	2175	**5.00**

Table 6.17 lists the results of experiments with EED for the old Marble sequence. For this sequence the robust combined optic flow method with EED gave the lowest AAE of all methods discussed here. With 5.00° the AAE was 0.02° lower than the AAE of the corresponding standard method, and 0.30° lower then the AAE of the Horn and Schunck method. The optic flow estimated with the combined method is shown in Figure 6.15 together with the combined optic flow estimation using the mixed homogeneous-biharmonic operator.

EED did not improve any results for the Yosemite sequences. Lowering the contrast parameter λ gave rise to oversegmentation of the motion field which yielded a worse AAE. Using a flow-driven anisotropic operator as in (3.21)-(3.22), where both eigenvalues of the diffusion tensor adapt to the local image structure, only improved the standard optic flow estimation for the new Marble sequence. Robust approximation however did become better.

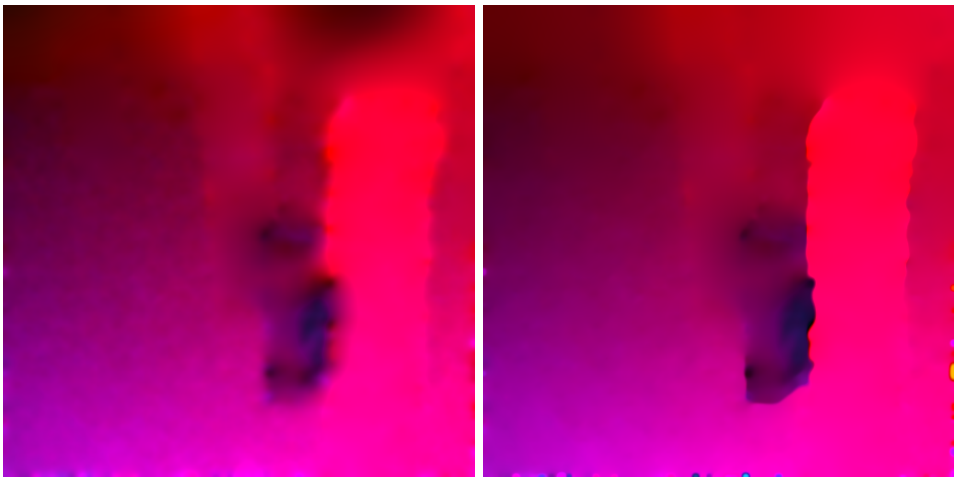

Figure 6.15: *The robust combined optic flow method for the old Marble sequence.* **Left:** *Mixed homogeneous-biharmonic operator, AAE = 5.05°.* **Right:** *EED, AAE = 5.00°.*

Image-driven Operator

EED seems to be the optimal choice to approximate a data set that has sharp discontinuities in the sample values and a more or less uniform spatial density. If a discontinuitiy in the sample values coincides with a discontinuity in the spatial density, most feature-based approximation methods seem to fail. Robust approximation performs bad in sparse data regions because there is not enough filling-in. This can be witnessed in the case of the Yosemite sequence with clouds, where even EED creates strong singularities in the cloud region. For this sequence the image-driven operator, as in (3.23)-(3.24), provides good results. Smoothing across image edges such as the horizon will be inhibited. Since the horizon coincides with a motion boundary, the SIFT displacements will be clustered in two populations with different types of motion. Even a very small SIFT feature density in the sky produced a reasonable estimate of the cloud motion.

Table 6.18: *Results for the Yosemite sequence with clouds. The estimation methods are the standard optic flow method, the interpolation of SIFT features with post-smoothing, the robust combined optic flow method. The smoothing operator is the homogeneous operator or the image-driven (ID) operator. The parameters σ, λ and the smoothness weight are optimized w.r.t. the AAE.*

method	smooth. operator	c	σ	λ	smooth. weight	AAE [°]
optic flow	hom.	-	1.33	-	453	7.17
	ID	-	1.33	0.8	$> 10^4$	**6.06**
interpolation	hom.	-	-	-	-	14.4
interpol. + post-smooth.		0.025	-	-	-	14.4
interpolation	*ID*	-	-	0.1	-	**6.82**
interpol. + post-smooth.		0.025	-	0.02	-	**6.50**
robust combined optic flow	hom.	0.990	1.33	-	550	7.14
	ID	0.999	1.33	0.8	$> 10^4$	**5.94**

Table 6.18 and Figure 6.16 show the results for image-driven interpolation for the Yosemite sequence with clouds. Pure interpolation of the SIFT displacements resulted in a flow estimation with an AAE of 6.82°. We chose a contrast parameter λ of 0.1, a noise scale σ of 1 and an integration scale ρ of 3. Subsequent smoothing of the interpolant reduced the AAE to 6.50°. Post-smoothing was done by applying the flow-driven operator

$$\operatorname{div}\left(g(\nabla u_\sigma \nabla u_\sigma^\top + \nabla v_\sigma \nabla v_\sigma^\top)\right)$$

with a different contrast parameter. This preserved the sharp edges from the initial interpolant. We can see from the plot of the u-component of the flow field that image-driven interpolation allows for a filling-in in the sky region. Both the motion of the clouds and the motion discontinuity at the horizon are estimated well. As a comparison, the standard image-driven optic flow estimation produced an AAE of 6.06°, but the ASE was higher than the ASE of the interpolant. The robust combined optic flow method gave an AAE of 5.94°. This is a small increase in quality over the standard method. Increasing the integration scale ρ gave slightly better AAEs for all methods, but the qualitative results remained the same.

The image-driven method did not perform well for the Marble sequences. It tends to give an oversegmentation of the flow field due to the strong texture in the images.

6.5.4 Selecting a Subset of SIFT displacements

In 6.5.1 we selected a subset of SIFT displacements with the help of ground truth information. We showed that this subset could improve the results of the Horn and Schunck method when we used their exact values in the optic flow estimation. This data set can most probably not be reproduced by any other means than by using ground truth. We can however try to approximate this subset by culling our data in another way.

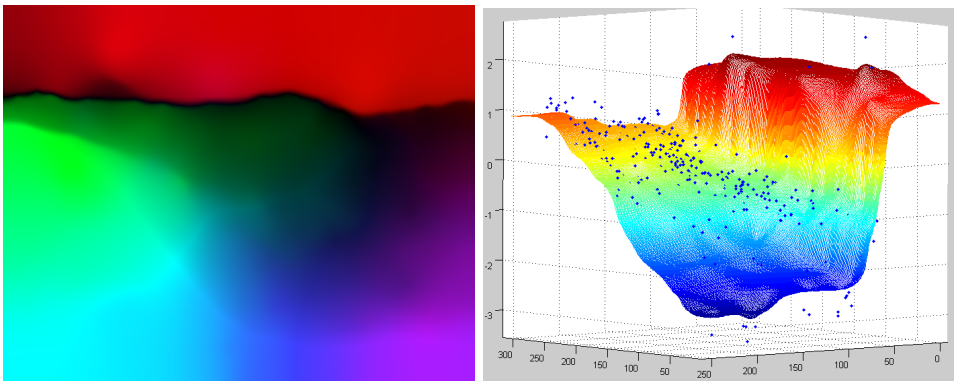

Figure 6.16: *Image-driven interpolation of the SIFT displacements with flow-driven post-smoothing for the Yosemite sequence with clouds.* **Left:** *The optic flow field, AAE = 6.50.* **Right:** *The u-component of the flow field.*

In 3.5 we presented two approaches for extracting a high quality subset from our SIFT displacements: a modified vector median filter (VMF) and the usage of an optic flow constraint (OFC). The VMF rejects displacements that do not respect a smoothness assumption, while the OFC rejects displacements that do not comply with a constancy assumption of certain image features. We will evaluate both approaches for two sets of SIFT displacements. One set has been obtained as in the previous experiments, namely with a distance ratio threshold between the closest and the second closest feature descriptor of 0.6. The other set has been obtained with a distance threshold of 0.8 and it typically contains more false matches. We did not remove large displacements prior to applying the modified VMF or the OFC. In this way we can asses their ability of handling severely outlying data.

In Table 6.19 we list the results of the experiments carried out with the modified VMF for the new Marble sequence. We tested interpolation with post-smoothing, the robust approximation scheme and the robust combined optic flow method with the EED operator. For the VMF we found an optimal radius R of 27 and a value for the parameter c of 0.98. In this way 2062 of 3518 SIFT-matches (not effective matches) were retained for a threshold of 0.6 and 2114 of 3596 for a threshold of 0.8.

We see from the table and from Figure 6.17 that the interpolation results improve substantially after applying the modified VMF to the raw data set. If we post-smooth the interpolant for the 0.6 threshold we end up with an optic flow estimation that has an AAE of 1.29°. This result is almost as good as the standard optic flow estimation with EED. Robust approximation improved only slightly after applying the VMF. We state that the overall effect of thinning out the raw data set by vector median filtering and interpolating the remaining subset of SIFT displacements is similar to performing robust approximation to the raw data set.

Robust combined optic flow after vector median filtering reduced the AAE from 1.25° to 1.23°. There was no difference in quality for the two distance thresholds.

Applying VMF was unfortunately not successful in selecting a higher quality subset for the remaining image sequences. SIFT produces isolated data points for the Yosemite sequence with clouds and the old Marble sequence. These isolated displacements are crucial for the accurate estimation of the flow field with a feature-based method but they were mostly discarded by our modified VMF. If the radius R was chosen large, isolated data points were discarded. If the radius was chosen small, no filtering took place. For isolated points it is difficult to tell if they are

Table 6.19: *Results for the new Marble sequence with and without VMF. The smoothness weight is optimized w.r.t. the AAE.*

thresh.	method	VMF	c	σ	λ	smooth. weight	AAE [°]
-	optic flow	-	-	0.564	0.004	$> 10^4$	1.26
0.6	interpolation	no	-	-	0.009	-	2.79
		yes	-	-	0.009	-	**1.65**
	interpolation + post-smoothing	no	0.003	-	0.009	-	1.76
		yes	0.003	-	0.009	-	**1.29**
	approximation	no	-	-	0.009	18.4	1.59
		yes	-	-	0.009	17.2	**1.49**
	robust combined optic flow	no	0.996	0.564	0.004	$> 10^4$	1.25
		yes	0.998	0.564	0.004	$> 10^4$	**1.23**
0.8	interpolation	no	-	-	0.009	-	4.63
		yes	-	-	0.009	-	**1.75**
	interpolation + post-smoothing	no	0.003	-	0.009	-	3.67
		yes	0.003	-	0.009	-	**1.36**
	approximation	no	-	-	0.009	17.7	1.63
		yes	-	-	0.009	19.1	**1.50**
	robust combined optic flow	no	0.996	0.564	0.004	$> 10^4$	1.25
		yes	0.999	0.564	0.004	$> 10^4$	**1.23**

outlying or not, due to a lack of local information. This renders the concept of outlier vague in a framework of scattered data approximation. A solution might be performing data clustering prior to performing subset selection.

The idea of making use of the optic flow constraint to select inlying data did not work out for any sequence. The set of SIFT displacements that was retained in this way was of poor quality and did not improve our estimation.

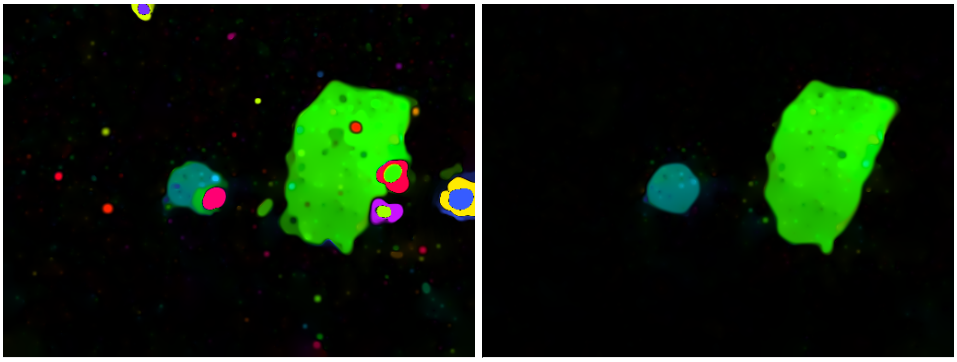

Figure 6.17: *The optic flow for the new Marble sequence estimated with EED interpolation from a set of SIFT displacements. The SIFT displacements were obtained with a distance ratio threshold of 0.8.* **Left:** *Without VMF, AAE = 4.63.* **Right:** *With bad displacements removed by VMF, AAE = 1.75.*

Chapter 7

Summary and Conclusions

Variational methods are among the most successful methods for computing the optic flow in an image sequence. They are highly accurate and do not encompass any hidden model assumptions. Variational methods establish a dense correspondence mapping between pixels in different images. On the other side there are local techniques that find a scattered set of correspondences between important locations of interest. Such feature-based methods typically produce a non-dense displacement field. Both methods are usually applied strictly separated and the use of the one in preference to the other depends on the purpose of the task.

The contribution of this work has been to bring together variational and feature-based approaches for optic flow estimation. We have accomplished the following:

- ◇ **Dense feature-based optic flow estimation.** We have developed a novel technique for calculating the optic flow field from a sparse displacement field. Our vector approximation scheme is comprised of a smoothness constraint and a similarity term that only exists in a discrete subset of the domain. The smoothness enters the model in the form of an elliptic operator. A variety of choices for the operator allows to achieve a specific kind of filling-in. A smoothness weight regulates the complexity of the data fitting model.
- ◇ **Combined optic flow estimation.** The close resemblance in formulation permits a straightforward incorporation of the feature-based method into the standard optic flow model. The continuous data constraint is supplemented with the discrete similarity term while smoothness is imposed by a common operator.
- ◇ **Experimental comparison.** We provided an experimental comparison of the three approaches by calculating the optic flow of different benchmark sequences. Simultaneously we evaluated the SIFT algorithm for motion estimation with small displacements. Strategies for removing outliers, as well as robust variants of our methods have been tested for their effectiveness.

Conclusions and Outlook

The outcome for feature-based methods with data from an actual feature extractor like SIFT was for some operators surprisingly good. Robust approximation with EED performed well if there were discontinuities in the SIFT displacement values and the locations of the displacements were uniformly distributed. If motion boundaries coincide with image edges and if there is not much texture present, image-driven operators can give fairly good results. In some cases the robust approximation gave a lower ASE than the corresponding standard optic flow method,

but the AAE was always higher. A natural remark is that the quality of the feature-based methods depends on the presence of point correspondences in a certain area. As the cloud region in the Yosemite sequence demonstrates, the estimation can be very good with a minimal number of data points. If this data lacks however, the result turns out to be unsatisfactory which is an inherent drawback of feature-based methods.

We can further conclude that a simple data constraint like the brightness constancy assumption outperforms a similarity constraint with SIFT displacements. The combination of both constraints, however, resulted in a marginal improvement of the AAE for all image sequences that were included in the tests. By using ground truth information we have shown that feature-based and combined optic flow methods can give far better results if the feature matching algorithm would be more accurate. The noise in the sparse displacement field that is inherent to SIFT prevents any major improvements of the standard optic flow methods for small displacements

Future work would include experiments with larger displacements. Variational optic flow methods typically perform less well for image sequences with displacements that are larger than several pixels. This is because the constancy assumptions are linearized, either in the model or in the discretization. We expect that a feature-based or combined method with SIFT will perform better for large displacement rates than for small displacements because the effects of the localization errors in SIFT will be less significant. We can also think about using other types of feature matching techniques, such as tracking algorithms. These algorithms are especially designed for keeping track of the same set of interest points throughout an image sequence, but they often make additional assumptions on the type of motion. Other optimizations would include more refined outlier detection, the use of spatiotemporal smoothing, and the consideration of other smoothing operators. In the latter case we could think about the Frobenius norm of the Hessian, which is closely related to the biharmonic operator. A smoothness constraint based on this norm with a suitable penalization might be able to prevent the over- and undershoots which are characteristic for biharmonic smoothing.

Bibliography

[1] J. Astola, P. Haavisto, and Y. Neuvo. Vector median filters. *Proc. IEEE*, 78(4):678–689, 1990.

[2] 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.

[3] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proceedings of the ninth European Conference on Computer Vision*, May 2006.

[4] M. Bertero, T. A. Poggio, and V. Torre. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889, August 1988.

[5] 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.

[6] 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.

[7] 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.

[8] 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*, volume 3951 of *Lecture Notes in Computer Science*, pages 471–483. Springer, Berlin, 2006.

[9] A. Bruhn. *Variational Optic Flow Computation, Accurate Modeling and Efficient Numerics*. PhD thesis, Universität des Saarlandes.

[10] 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.

[11] 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.

[12] 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*, 70(3), December 2006.

[13] 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.

[14] 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.

[15] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[16] 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.

[17] D. Chetverikov and J. Verestóy. Tracking feature points: A new algorithm. *In Proc. International Conf. on Pattern Recognition*, pages 1436–1438, 1998.

[18] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint.* MIT Press, Cambridge, MA, 1993.

[19] 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.

[20] I. Galić, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, and H. P. Seidel. Towards PDE-sased image compression. *Proc. VLSM 2005, Lecture Notes in Computer Science*, 3752(1):37–48, 2005.

[21] I. M. Gelfand and S. V. Fomin. *Calculus of Variations.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, revised english edition, 1963.

[22] C. G. Harris and M. Stephens. A combined corner and edge detector. In *Proc. Fouth Alvey Vision Conference*, pages 147–152, Manchester, England, August 1988.

[23] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, Cambridge, UK, 2000.

[24] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer, 2001.

[25] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[26] Y. Ke and R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2:II–506–II–513 Vol.2, 2004.

[27] T. Lindeberg. *Scale-Space Theory in Computer Vision.* Kluwer, Boston, 1994.

[28] D. G. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2004.

[29] 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.

[30] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London, Series B*, 207:187–217, 1980.

[31] 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.

[32] E. Mémin and P. Pérez. Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision*, 46(2):129–155, 2002.

[33] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 257–263, June 2003.

[34] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[35] 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.

[36] 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*, 67(2):141–158, April 2006.

[37] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.

[38] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection.* John Wiley & Sons, Inc., New York, NY, USA, 1987.

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

[40] C. V. Stewart. Robust parameter estimation in computer vision. *SIAM Review*, 41(3):513–537, 1999.

[41] J. Weickert. *Anisotropic Diffusion in Image Processing.* PhD thesis, Department of Mathematics, University of Kaiserslautern, Germany, January 1996. Revised and extended version published by Teubner, Stuttgart, Germany, 1998.

[42] 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.

[43] 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.

[44] J. Weickert and M. Welk. tensor field interpolation with PDEs. *J. Weickert, H. Hagen (Eds.): Visualization and Processing of Tensor Fields*, 34:315–325, 2006.

[45] A. P. Witkin. Scale-space filtering. In *Proc. Eighth International Joint Conference on Artificial Intelligence*, volume 2, pages 945–951, Karlsruhe, West Germany, August 1983.

[46] D. M. Young. *Iterative Solution of Large Linear Systems.* Academic Press, New York, 1971.

[47] Zhengyou Zhang, Rachid Deriche, Olivier D. Faugeras, and Quang-Tuan Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, 1995.