Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

Anisotropic Selection Schemes for Order-Adaptive Variational Optical Flow Methods

Lukas Francesco Mehl

Course of Study:	Informatik
Examiner:	Prof. Dr.-Ing. Andrés Bruhn
Supervisor:	Prof. Dr.-Ing. Andrés Bruhn, Dr. Daniel Maurer, Michael Stoll, M.Sc. M.Sc. B.Sc.
Commenced:	October 2, 2019
Completed:	March 19, 2020

Abstract

Variational optical flow estimation, i.e. the estimation of motion information from image sequences using continuous optimization approaches, plays an important role in computer vision. Recently, the order-adaptive regularization approach by Maurer *et al.* [MSB17b] has shown to yield a performance improvement by not restricting itself to first- or second-order regularization but selecting the optimal order locally adaptive. Although the regularizers included in their method are anisotropic, i.e. adaptive to local image directions, the order selection is not. The selection scheme is restricted to always choosing the same order for different local directions, which is a shortcoming that will be addressed in this thesis. After providing an introduction into the topic and a detailed derivation, minimization and discretization of the method by Maurer *et al.*, this thesis presents the model for an order-adaptive approach that includes an anisotropic order selection scheme. The newly presented method selects the regularization order locally adaptive and individually for two directions corresponding to image structures. Additionally, by deriving the minimization of the novel method, it is shown that the same optimization as in the work of Maurer *et al.* [MSB17b] is possible. Finally, the approaches are compared using an evaluation on recent benchmarks and an analysis of the resulting order maps is given. While the results do not show an improvement on the benchmarks, experiments including the ground truth data show that the idea has potential.

Zusammenfassung

Variationsansätze zur Berechnung des optischen Flusses, d.h. zur Schätzung von Bewegungsinformationen aus Bildsequenzen mit Hilfe von kontinuierlichen Optimierungsansätzen, spielen eine wichtige Rolle im Maschinensehen. Der kürzlich vorgestellte ordnungsadaptive Regularisierungsansatz von Maurer *et al.* [MSB17b] konnte eine Qualitätsverbesserung dadurch erreichen, dass er nicht auf die Regularisierung erster oder zweiter Ordnung beschränkt ist, sondern die optimale Ordnung lokal adaptiv auswählt. Obwohl die in deren vorgestelltem Verfahren enthaltenen Regularisierer anisotrop, d.h. adaptiv zu lokalen Bildrichtungen, sind, ist die Auswahl der Ordnung das nicht. Das Selektionsschema ist darauf beschränkt, für unterschiedliche lokale Richtungen immer die gleiche Ordnung zu wählen. Dies ist ein Nachteil, der in der vorliegenden Arbeit behoben werden soll. Nach einer Einführung in das Thema und einer detaillierten Herleitung, Minimierung und Diskretisierung der Methode von Maurer *et al.* wird in dieser Masterarbeit das Modell eines ordnungsadaptiven Ansatzes vorgestellt, der ein anisotropes Ordnungsselektionsschema beinhaltet. Das neu vorgestellte Verfahren wählt die Regularisierungsordnung lokal adaptiv und individuell für zwei Richtungen aus, die Bildstrukturen entsprechen. Zusätzlich wird durch die Herleitung der Minimierung des neuen Verfahrens gezeigt, dass die gleiche Optimierung wie in der Arbeit von Maurer *et al.* [MSB17b] möglich ist. Abschließend werden die Ansätze anhand einer Auswertung auf Basis aktueller Benchmarks verglichen und eine Analyse der resultierenden Ordnungskarten durchgeführt. Die Ergebnisse zeigen zwar keine Verbesserung auf den Benchmarks, aber Experimente mit den in den Trainingsdaten gegebenen echten Flussfeldern zeigen, dass die Idee Potenzial hat.

Contents

1 Introduction

1.1 Motivation

Optical flow estimation is a central problem in computer vision. Given two consecutive images of an image sequence, a field of displacement vectors corresponding to the displayed motion is to be determined.

Calculating the optical flow field provides several applications in different domains. The optical flow can provide a good basis for object tracking, e.g. used by Lefaix *et al.* [LMB02] to detect obstacles based on the motion estimation or for point tracking as proposed by Sundaram *et al.* [SBK10]. Moreover, the optical flow can be used together with a depth reconstruction to detect and segment moving objects in an automotive context [KVR+09], for vehicle detection [KPL11] and for advanced driver assistance systems [OS14]. Another application lies in the field of video manipulation. Optical flow is needed for motion-compensated frame interpolation used e.g. for video retiming, restoration or temporal upsampling [RRBW12]. Additionally, optical flow can be used for video compression [SP01] and has been proposed for markerless motion capture [GVWT13].

For nearly forty years, variational methods have been developed in order to solve this problem. Such methods are global minimization approaches that compute the desired motion as the minimizer of a specific energy functional. This functional is central in a way that it models the assumptions a method imposes on the data. Over the last decades, a typical two-part structure of the energy functional has evolved. While the *data term* models constancy assumptions of corresponding image structures of the two frames, the *smoothness term*, or sometimes called *regularization term* formulates a smoothness assumption on the resulting vector field. The incorporation of a smoothness assumption additionally solves the problem that the results of the data term constancy assumptions are non-unique, a problem known as the *aperture problem*. Additionally, it helps achieve results in homogeneous and occluded image areas due to the *filling-in effect* – the propagation of information from the surrounding neighborhood.

When choosing a smoothness term, an important choice is the selection of the regularization order. While first-order regularization penalizes the first derivative of the resulting flow field and enforces the estimation of a constant or piecewise constant motion field, second-order regularization penalizes the second derivative and thus favors a flow field with linear changes. Figure 1.1 visualizes these two typical cases. The rows are showing Sequences 1 and 3 of the classroom dataset, with the first two columns showing the first and second image. The third column depicts the ground truth flow using the color visualization by Bruhn [Bru06] and the last column shows plots of the two flow components for a single scanline through the image. It becomes obvious that for the first sequence, which contains only fronto-parallel motion, first-order regularization is a good choice, since the resulting flow is piecewise constant. For the second sequence, where the camera moves forward, a second-order penalizer is favorable, since the flow is piecewise linear.

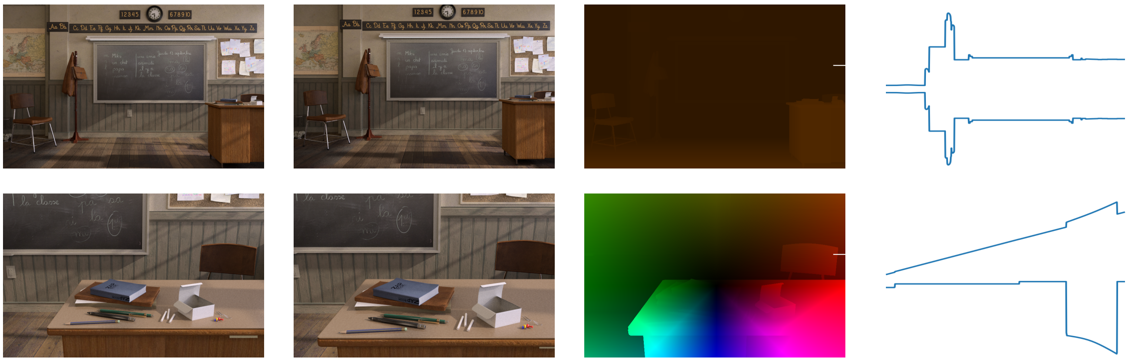


Figure 1.1: Example cases for first- and second-order regularization. *From left to right:* First and second image, resulting flow field color-coded using the strategy by Bruhn [Bru06], plot of the two flow components u and v for a scanline of the ground truth flow (marked in the flow visualization). *From top to bottom:* Sequence 1 and 3 of the classroom dataset.

Numerous approaches have been proposed in the literature that make use of first-order regularization, starting from the seminal method by Horn and Schunck [HS81] and including many of its improved variants [BA96; BBPW04; NE86; WTP+09; ZBW11]. Since then, more recent approaches have been proposed that restrict themselves to a second order setting [BDB13; DSV+14; HWS+13; KC13; TPCB08]. In order to overcome this choice between regularization approaches, an order-adaptive method was presented by Maurer *et al.* [MSB17b]. Central to their idea is the computation of an order selection map that locally decides on the optimal regularization order, an approach that provides good results in all recent benchmarks. Inside their approach they make use of an anisotropic first- and second-order regularizer from which they adaptively select the best one. Anisotropic approaches have proven to be beneficial in many cases [MSV+17] and allow for a precise regularization depending on the direction of underlying image structure. However, although the regularizers used by Maurer *et al.* [MSB17b] are anisotropic, the order selection scheme they propose is still isotropic.

In order to overcome this drawback of having to select the same order in each anisotropic direction, in this thesis an extension to the method of Maurer *et al.* [MSB17b] will be presented. The novel method introduces an anisotropic order selection scheme that allows for an adaptive order selection for each local direction.

1.2 Related Work

In the recent years, several variations of regularization terms were proposed, concerning the penalized order, robust non-linear terms, the introduction of anisotropy and adaptive approaches. Moreover, the best-performing ideas are embedded in the state-of-the-art pipeline approaches. All these aspects will be discussed in the following.

Single-Order Regularization The concept of selecting a single order for the regularization term applies to a vast majority of variational optical flow methods. While the traditional approach of first-order-only regularization dates back to the method of Horn and Schunck [HS81], it is still included in later approaches e.g. by Brox *et al.* [BBPW04] or Zimmer *et al.* [ZBW11]. More recent approaches make use of a second-order regularization, which can be achieved in two ways. The most straight-forward way is a direct penalization of the second derivative, e.g. in terms of penalizing the Frobenius norm of the Hessian as used by Demetz *et al.* [DSV+14] or the Laplacian as used by Chan *et al.* [CMM00]. On the other hand, several authors make use of an indirect approach, where the main idea is to introduce an auxiliary field to approximate the first-order derivative of the flow. This field in turn is then penalized using a first-order smoothness assumption. This way, the regularization is not applied directly to the flow field, but to the coupling variable, which explains why this approach is sometimes referred to as *coupling model*. The approach was first introduced as Total Generalized Variation (TGV) by Bredies *et al.* [BKP10] and later used for general optical flow estimation by Braux-Zin *et al.* [BDB13] and with a focus on mechanics by Hewer *et al.* [HWS+13]. Additionally a non-local TGV variant was introduced by Ranftl *et al.* [RBP14].

Anisotropic Regularization The integration of directional information into the regularization term has been around for a long time. It was introduced by Nagel and Enkelmann [NE86], who referred to it as *oriented smoothness*. Since then, several variations of *image-driven regularization* have been proposed. The method of Werlberger *et al.* [WTP+09] paired a Total Variation model with directional information. Zimmer *et al.* [ZBW11] harmonized the ideas of image- and flow-driven regularization by introducing the *regularization tensor*, from which the directional information is retrieved. In a dual way, the regularization tensor directly relates to the data term constraints and the motion tensor. Based on the TGV approach, Kuschk and Cremers [KC13] proposed an anisotropic structure-adaptive variant for stereo-reconstruction. Finally, Hafner *et al.* [HSW15] proposed a method introducing anisotropy into the regularization as well as the coupling term of a TGV model in the context of focus fusion.

All approaches listed so far are based on a specific regularization order with different usages of anisotropic regularization. Maurer *et al.* [MSV+17] present a good comparison of these two dimensions.

Order-Adaptive Regularization Selecting a fixed regularization order yields good results in one benchmark, but lets the method perform worse on another, thus a robust “one-fits-all” approach is lacking. Only the coupling models as described above implicitly already perform an order-adaptive regularization approach: The auxiliary variables are penalized with a first-order method, but – since they already correspond to the first derivative of the flow – this penalization can also be regarded as second-order. A first approach in the direction of explicit order adaptation is provided by Lenzen *et al.* [LBL13] in the context of denoising. They use a first- and second-order regularization term which they adaptively switch using local image information. In the context of optical flow estimation, the first order-adaptive approach was proposed by Maurer *et al.* [MSB17b]. They use two anisotropic regularizers of first and second order where the latter is given as a coupling approach and they additionally propose order selection schemes that switch between the two.

Pipeline Approaches The latest state-of-the-art methods make use of pipelines combining several steps, where the last step strives to refine a previously estimated flow field using variational optical flow estimation. A well-performing example of this is the method by Maurer *et al.* [MSB17a] which directly makes use of the isotropic order selection approach. Their refinement procedure is then used in some of the latest methods that show a high performance on current benchmarks such as the Structure-from-Motion-Aware PatchMatch approach by Maurer *et al.* [MMGB18] or the ProFlow method by Maurer and Bruhn [MB18]. Additionally, the refinement procedure making use of the order-adaptive scheme is extended by Maurer *et al.* [MSB18] to a multi-frame setup. With all these applications present, an improvement of the order-adaptive method would directly contribute to the improvement of several state-of-the-art methods.

1.3 Thesis Organization

The remaining thesis is organized as follows: Chapter 2 explains the background needed in order to derive more ambitious methods. Afterwards, Chapter 3 gives a detailed description of the baseline method by Maurer *et al.* [MSB17b]. This chapter provides not only the derivation of the model and the minimizing equations, but also a thorough derivation of the discretization. With this as a base, the newly developed anisotropic order selection method is introduced in Chapter 4. Chapter 5 then analyzes and evaluates the proposed selection scheme and compares it to its isotropic counterpart. In the end, Chapter 6 summarizes the main points and gives an outlook to possible future approaches.

2 Background

2.1 Notation

In the coming chapters, the following notational settings will be used. Vectors and tensors are denoted by bold lower and upper case symbols. In order to have a more compact notation, derivatives will frequently be abbreviated by a subscript, i.e. $f_x = \frac{\partial f}{\partial x}$. Additionally, stencils are denoted as $\mathcal{S}$.

The gradient operator ∇ is defined as the vector of partial gradients. If not denoted otherwise, throughout this thesis it is defined as the two-dimensional spatial gradient with derivatives w.r.t. x and y. It is given by $\nabla f = (f_x, f_y)^\top$. Similarly, the Laplacian Δ is also defined for the two spatial dimensions. Thus, its definition is $\Delta f = f_{xx} + f_{yy}$.

2.2 Variational Methods for Optical Flow

Optical Flow The optical flow is the motion field between two consecutive frames of an image sequence. The goal of optical flow estimation methods is to create a dense 2D vector field by determining the displacements of pixel correspondences from the reference frame to the second frame. Optical flow originates from camera or object motion in a 3D scene. The projection of this 3D motion onto the 2D image plane yields the optical flow.

Let $f(\mathbf{x}, t)$ be an image sequence, where $\mathbf{x} = (x, y)^\top \in \Omega$ denotes the spatial location on the image domain Ω and $t \in \mathbb{R}_0^+$ the time step. Then the optical flow $\mathbf{u}(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}))^\top \in \mathbb{R}^2$ describes for every location $\mathbf{x}$ the motion $\mathbf{u}$ in image space from the reference time step t to $t+1$. In some cases, it is useful to introduce a notation, where the optical flow vector is extended by a one, denoting the temporal difference with one time step. In those cases, the optical flow definition $\mathbf{w}(\mathbf{x}) = (\mathbf{u}(\mathbf{x})^\top, 1)^\top = (u(\mathbf{x}), v(\mathbf{x}), 1)^\top$ is used. Throughout this thesis, both formulations will be used interchangeably.

Variational Methods for Optical Flow Variational methods calculate minimizers of functionals. A functional is – compared to a function which is a finite-dimensional mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$ – defined on an infinite-dimensional domain of mappings [Kie18]. It can be interpreted as a mapping from a set of possible functions to a scalar value, thus the input of a functional is a function and the output a scalar value.

Using a continuous formulation of the image sequence f and the optical flow $\mathbf{u}$, the computation of the optical flow can be formulated as a minimization problem based on an energy functional: The energy functional $E(\mathbf{u})$ maps every possible optical flow $\mathbf{u}$ to a scalar energy value E. This allows to transparently model assumptions on the desired solution and assess the quality of a given solution with respect to the assumptions. The energy can be interpreted as a cost where deviations from the assumptions are penalized with a higher cost. The assumptions are global, i.e. they must hold on the entire image domain. Thus, for some model term F, an energy functional usually has the form

$$E(\mathbf{u}) = \int_\Omega F \, d\mathbf{x} \; . \tag{2.1}$$

The optimal result can then be found by determining the minimizer of this energy functional.

2.2.1 Euler-Lagrange Equations

In order to find this minimizer, the Euler-Lagrange equations can be derived as necessary conditions. Analogously to a continuous function, where the necessary condition to find the minimum is that the first derivative must vanish, the Euler-Lagrange equations are the condition that must hold for the minimizing function.

The Euler-Lagrange equations are a system of differential equations that were developed independently by Euler and Lagrange around 1755 [Kie18]. In the following, they will be given in their general form, first for a one-dimensional case and afterwards for a multivariate case.

One-dimensional Case A one-dimensional energy functional $E(u)$ that integrates over some domain $[a; b]$ and where the argument of its integral only depends on x, $u(x)$ and its first derivative $u'(x)$ has the form

$$E(u) = \int_a^b F(x, u, u') \, dx \; . \tag{2.2}$$

Then, the necessary condition for a minimizer is given by

$$F_u - \frac{d}{dx} F_{u'} = 0 \; , \tag{2.3}$$

where additionally the condition $F_{u'} = 0$ must hold for the domain boundaries where $x = a$ and where $x = b$. These additional constraints are thus called *boundary conditions*.

Multivariate Case Let an energy functional for a function $\mathbf{u}$, which is defined on a 2D domain Ω with $\mathbf{x} \in \Omega \subset \mathbb{R}^2$ and that has a 2D co-domain with $\mathbf{u}(\mathbf{x}) \in \mathbb{R}^2$, have the form

$$E(\mathbf{u}) = \int_\Omega F(x, y, u, v, u_x, u_y, v_x, v_y) \, d\mathbf{x} \; . \tag{2.4}$$

Then, its necessary conditions are given by the two Euler-Lagrange equations

$$F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} = 0 \; , \tag{2.5}$$

$$F_v - \frac{\partial}{\partial x} F_{v_x} - \frac{\partial}{\partial y} F_{v_y} = 0 \; . \tag{2.6}$$

Additionally, the boundary conditions must hold, which can be defined using the vector $\mathbf{n}$ pointing across the image domain boundary. They read

$$\mathbf{n}^\top \begin{pmatrix} F_{u_x} \\ F_{u_y} \end{pmatrix} = 0, \quad \mathbf{n}^\top \begin{pmatrix} F_{v_x} \\ F_{v_y} \end{pmatrix} = 0 \; . \tag{2.7}$$

2.3 The Method of Horn and Schunck

2.3.1 Model

Variational optical flow methods usually consist of a two-part energy functional that comprises a *data term* D and a *smoothness term* S. Both terms are integrated over the image domain Ω and balanced with a parameter α, yielding the structure

$$E(\mathbf{u}) = \int_\Omega D + \alpha \cdot S \, d\mathbf{x} \; . \tag{2.8}$$

In the following, the two terms will be presented with two possible choices for each of them leading to the well-known model of Horn and Schunck [HS81].

Data Term In order to model the correspondence between the two images, the data term includes constancy assumptions on the image data. A simple example would be the *brightness constancy assumption*, where it is assumed that the image brightness is the same at any original pixel position in the first image $f(\mathbf{x}, t)$ and its corresponding displaced pixel position in the second image $f(\mathbf{x} + \mathbf{u}, t + 1)$:

$$f(\mathbf{x}, t) = f(\mathbf{x} + \mathbf{u}, t + 1) \; . \tag{2.9}$$

When introducing a Taylor expansion of f around the point $(\mathbf{x}, t)$ and evaluating it at $(\mathbf{x} + \mathbf{u}, t + 1)$, the constancy assumption can be approximated by the linearized form:

$$f_x u + f_y v + f_t \approx 0 \; . \tag{2.10}$$

This approximation is only valid if u and v are small or if f varies slowly.

Smoothness Term The smoothness term incorporates an additional assumption on the solution: The solution is assumed smooth, i.e. it spatially varies only slowly. This can be modeled by imposing a quadratic penalizer on the spatial flow gradient and enforcing the following term to be small for all pixel locations $\mathbf{x}$:

$$|\nabla u|^2 + |\nabla v|^2 \; . \tag{2.11}$$

Figure 2.1 visualizes three example flow fields and their corresponding energy according to Equation (2.11). This shows that the smoothness term is a measure of the homogeneity of the flow field: If a functional consisted only of the smoothness term, the minimal energy with a value of zero would be achieved for a constant flow field that has the same displacement vector for every spatial position.

Energy Functional In order to find a compromise between the two assumptions, they are considered jointly in an energy functional. When combining the squared linearized brightness constancy assumption from Equation (2.10) and the quadratic penalization of the spatial flow gradient from Equation (2.11), the result is the energy functional for the method of Horn and Schunck [HS81]:

$$E(\mathbf{u}) = \int_\Omega \Big((f_x u + f_y v + f_t)^2 + \alpha \cdot \big(|\nabla u|^2 + |\nabla v|^2 \big) \Big) \, d\mathbf{x} \; . \tag{2.12}$$

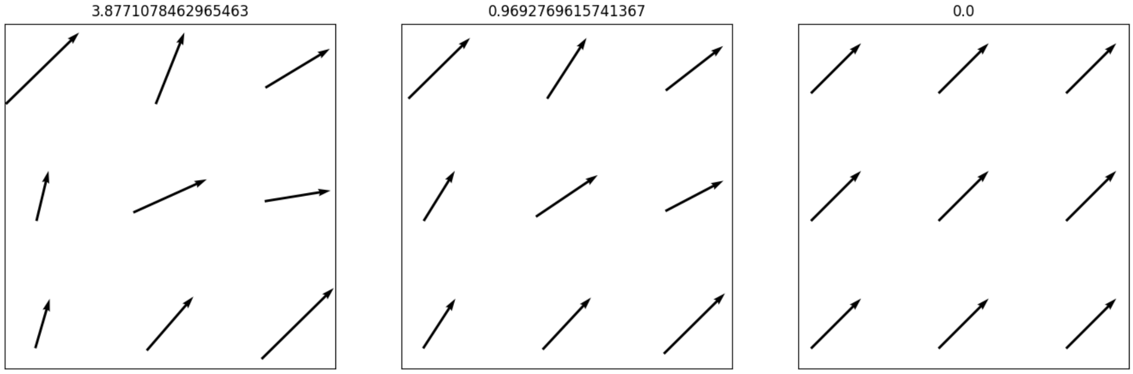


Figure 2.1: Three example flow fields of increasing homogeneity with the corresponding energy as described in Equation (2.11).

The parameter α balances the two assumptions: A larger value of α corresponds to a higher emphasis of the smoothness term, a smaller one to more weight on the data term. In the extreme case, a value of infinity will ignore the data term, such that the minimizing function comes down to a constant flow (see Figure 2.1).

In order to simplify the formulation, the motion tensor notation $\mathbf{J}$ [Bru06] can be introduced. The motion tensor is defined as

$$\mathbf{J} = \begin{pmatrix} f_x \\ f_y \\ f_t \end{pmatrix} \cdot \begin{pmatrix} f_x \\ f_y \\ f_t \end{pmatrix}^\top . \tag{2.13}$$

Equation (2.12) can then be rewritten in a more compact form which will be useful later. For this, the extended flow abbreviation $\mathbf{w} = (\mathbf{u}^\top, 1) = (u, v, 1)^\top$ is used and the energy functional reads

$$E(\mathbf{u}) = \int_\Omega \left(\mathbf{w}^\top \mathbf{J} \mathbf{w} + \alpha \cdot \left(|\nabla u|^2 + |\nabla v|^2 \right) \right) \, d\mathbf{x} \, . \tag{2.14}$$

The data term as well as the smoothness term of the energy functional are squared, which makes them strictly convex. Further, the positive combination of two strictly convex functions using α and the integration does not change this, which makes the complete energy functional of Horn and Schunck [HS81] strictly convex. Additionally, Weickert and Schnörr [WS01] proved that the method is well-posed, i.e. a unique solution exists that continuously depends on the input data. This simplifies the minimization, which will be discussed in the next section.

2.3.2 Minimization

The minimizer of a functional like in Equation (2.12) is a function and it is exactly the function $\mathbf{u}$ for which $E(\mathbf{u})$ is minimal. For the method of Horn and Schunck, the minimality conditions are given by the Euler-Lagrange equations as introduced in Equations (2.5) and (2.6) and come down to

$$f_x(f_x u + f_y v + f_t) - \alpha \, \Delta u = 0 \, , \tag{2.15}$$

$$f_y(f_x u + f_y v + f_t) - \alpha \, \Delta v = 0 \, , \tag{2.16}$$

with the boundary conditions

$$\mathbf{n}^\top \nabla u = 0, \quad \mathbf{n}^\top \nabla v = 0 \; . \tag{2.17}$$

Again making use of the motion tensor, Equations (2.15) and (2.16) are simplified to

$$\mathbf{J}_{11} \cdot u + \mathbf{J}_{12} \cdot v + \mathbf{J}_{13} - \alpha \, \Delta u = 0 \; , \tag{2.18}$$

$$\mathbf{J}_{21} \cdot u + \mathbf{J}_{22} \cdot v + \mathbf{J}_{23} - \alpha \, \Delta v = 0 \; . \tag{2.19}$$

2.3.3 Discretization

In order to discretize Equations (2.18) and (2.19), their components need to be discretized: The image domain Ω is subdivided into a grid with spacings h_x and h_y. The grid points are then given by the position $(i \cdot h_x, j \cdot h_y)$ with $(i, j) \in [0, M] \times [0, N]$. The wanted displacement vector $(u, v)^\top$ is then given by

$$u_{ij} := u(i \cdot h_x, j \cdot h_y) \; , \tag{2.20}$$

$$v_{ij} := v(i \cdot h_x, j \cdot h_y) \tag{2.21}$$

for all i, j. The image sequence is discretized in the same manner with $f_{i,j,t} := f(i \cdot h_x, j \cdot h_y, t)$. The spatial derivatives f_x and f_y can be approximated by central differences and the time derivative f_t by a forward difference. The temporal distance h_t between the frames is assumed to be 1. The derivative approximations then read

$$[f_x]_{ij} = \frac{f(i+1, j, t) - f(i-1, j, t)}{2 \cdot h_x} \; , \tag{2.22}$$

$$[f_y]_{ij} = \frac{f(i, j+1, t) - f(i, j-1, t)}{2 \cdot h_y} \; , \tag{2.23}$$

$$[f_t]_{ij} = f(i, j, t+1) - f(i, j, t) \; . \tag{2.24}$$

Using these definitions, the motion tensor entries are discretized as $[\mathbf{J}]_{ij}$. To discretize the Laplacian Δ, a set of two nested central differences can be used which results in

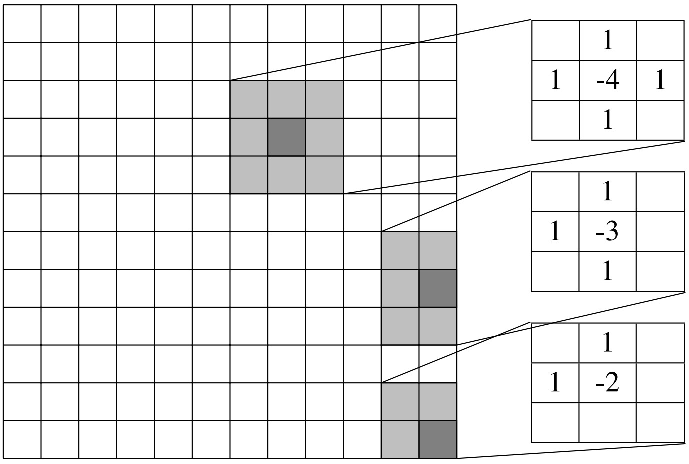
$$[\Delta u]_{ij} = \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{u_{\tilde{i},\tilde{j}} - u_{i,j}}{h_l^2} \; , \tag{2.25}$$

$$[\Delta v]_{ij} = \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{v_{\tilde{i},\tilde{j}} - v_{i,j}}{h_l^2} \; , \tag{2.26}$$

where $\mathcal{N}_l(i, j)$ describes the neighborhood of (i, j) in direction l. At the image boundary, only the neighbor inside the image domain is considered, which directly integrates the boundary conditions as described by Equation (2.17). The completely discretized Equations (2.18) and (2.19) then become for all i, j

$$0 = [\mathbf{J}_{11}]_{ij} \cdot u_{ij} + [\mathbf{J}_{12}]_{ij} \cdot v_{ij} + [\mathbf{J}_{13}]_{ij} - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{u_{\tilde{i},\tilde{j}} - u_{i,j}}{h_l^2} \; , \tag{2.27}$$

$$0 = [\mathbf{J}_{21}]_{ij} \cdot u_{ij} + [\mathbf{J}_{22}]_{ij} \cdot v_{ij} + [\mathbf{J}_{23}]_{ij} - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{v_{\tilde{i},\tilde{j}} - v_{i,j}}{h_l^2} \; . \tag{2.28}$$



	1	
1	-4	1
	1	

	1	
1	-3	
	1	

	1	
1	-2	

Figure 2.2: Spatially dependent stencil for the discretized Laplace operator. Example values for a central, border and corner pixel position. The grid sizes are assumed to be $h_x = h_y = 1$.

This system of equations is linear w.r.t. all u_{ij} and v_{ij}. It contains $2 \cdot M \cdot N$ equations for the same number of unknowns.

The discretization of the Laplacian can also be written in stencil notation. A stencil is defined as a weighted sum of the local neighborhood. For a pixel position (i, j) where all four neighbors exist, the corresponding stencil is

$$\mathcal{S}_{i,j} = \begin{array}{|c|c|c|} \hline & 1/h_y^2 & \\ \hline 1/h_x^2 & -2/h_x^2 - 2/h_y^2 & 1/h_x^2 \\ \hline & 1/h_y^2 & \\ \hline \end{array} \; . \tag{2.29}$$

If an indicator function χ is integrated, the stencil notation can be given including the boundary conditions. The indicator function is defined as

$$\chi_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in \Omega \\ 0 & \text{else} \end{cases} \; . \tag{2.30}$$

Then the stencil becomes

$$\mathcal{S}_{i,j} = \begin{array}{|c|c|c|} \hline & \frac{\chi_{i,j+1}}{h_y^2} & \\ \hline \frac{\chi_{i-1,j}}{h_x^2} & -\frac{\chi_{i-1,j}}{h_x^2} - \frac{\chi_{i+1,j}}{h_x^2} - \frac{\chi_{i,j-1}}{h_y^2} - \frac{\chi_{i,j+1}}{h_y^2} & \frac{\chi_{i+1,j}}{h_x^2} \\ \hline & \frac{\chi_{i,j-1}}{h_y^2} & \\ \hline \end{array} \; . \tag{2.31}$$

Figure 2.2 visualizes the resulting spatially variant stencil. Three example values for a central, border and corner position are shown. For the grid sizes, $h_x = h_y = 1$ is assumed.

Finally, using the stencil $\mathcal{S}$, Equations (2.27) and (2.28) can be written as:

$$0 = [\mathbf{J}_{11}]_{ij} \cdot u_{ij} + [\mathbf{J}_{12}]_{ij} \cdot v_{ij} + [\mathbf{J}_{13}]_{ij} - \alpha \sum\sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \mathcal{S}_{i,j,\tilde{i},\tilde{j}} \cdot u_{i,j} \; , \tag{2.32}$$

$$0 = [\mathbf{J}_{21}]_{ij} \cdot u_{ij} + [\mathbf{J}_{22}]_{ij} \cdot v_{ij} + [\mathbf{J}_{23}]_{ij} - \alpha \sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}}\sum \mathcal{S}_{i,j,\tilde{i},\tilde{j}} \cdot v_{i,j} \, . \tag{2.33}$$

The stencil notation introduced here will be used in later sections to derive more complex discretizations.

2.3.4 Structure of the Linear System of Equations

The system of equations given by Equations (2.27) and (2.28) is linear and thus can be written in the matrix form $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$, where $\mathbf{x}$ is a flattened vector of size $2 \cdot M \cdot N$ in which all unknowns u_{ij} and v_{ij} are stacked row-wise. The vector $\mathbf{b}$ of the same size contains all parts that are not multiplied with the unknowns and corresponds to the right-hand side of the equation. It contains in a stacked form all $[\mathbf{J}_{13}]_{ij}$ and $[\mathbf{J}_{23}]_{ij}$. The two vectors are given by

$$\mathbf{x} = \left(\begin{array}{c} u_{0,0} \\ \vdots \\ u_{N,M} \\ \hline v_{0,0} \\ \vdots \\ v_{N,M} \end{array}\right) , \qquad \mathbf{b} = \left(\begin{array}{c} -[\mathbf{J}_{13}]_{0,0} \\ \vdots \\ -[\mathbf{J}_{13}]_{N,M} \\ \hline -[\mathbf{J}_{23}]_{0,0} \\ \vdots \\ -[\mathbf{J}_{23}]_{N,M} \end{array}\right) . \tag{2.34}$$

Matrix $\mathbf{A}$ has size $(2 \cdot M \cdot N)^2$ and contains the rest of the Equations (2.27) and (2.28). It is symmetric and positive semi-definite, as proven by Sundaram *et al.* [SBK10], and given by

$$\mathbf{A} = \left(\begin{array}{c|c} \mathbf{diag}([\mathbf{J}_{11}]) & \mathbf{diag}([\mathbf{J}_{12}]) \\ \hline \mathbf{diag}([\mathbf{J}_{12}]) & \mathbf{diag}([\mathbf{J}_{22}]) \end{array}\right) - \alpha \cdot \left(\begin{array}{c|c} \langle \mathcal{S} \rangle & \mathbf{0} \\ \hline \mathbf{0} & \langle \mathcal{S} \rangle \end{array}\right) . \tag{2.35}$$

$\mathbf{A}$ is made up from 4 blocks of size $(M \cdot N)^2$. The first block relates to the coupling of the variables u_{ij} with themselves. It contains all motion tensor entries $[\mathbf{J}_{11}]_{ij}$ on the diagonal:

$$\mathbf{diag}([\mathbf{J}_{11}]) = \begin{pmatrix} [\mathbf{J}_{11}]_{0,0} & & 0 \\ & \ddots & \\ 0 & & [\mathbf{J}_{11}]_{N,M} \end{pmatrix} . \tag{2.36}$$

Similarly, $\mathbf{diag}([\mathbf{J}_{22}])$ corresponds to coupling of the variables v_{ij} with themselves and the block $\mathbf{diag}([\mathbf{J}_{12}])$, which appears twice, couples u and v.

In order to reformulate the stencil $\mathcal{S}_{i,j}$ to fit the matrix notation, it needs to be transformed to image space and unwrapped, which will be denoted as $\langle \mathcal{S}_{i,j} \rangle$. To compute it, a zero-initialized grid of size $M \times N$ is used with the stencil positioned at (i, j). Then, this 2D matrix is flattened to a 1D row

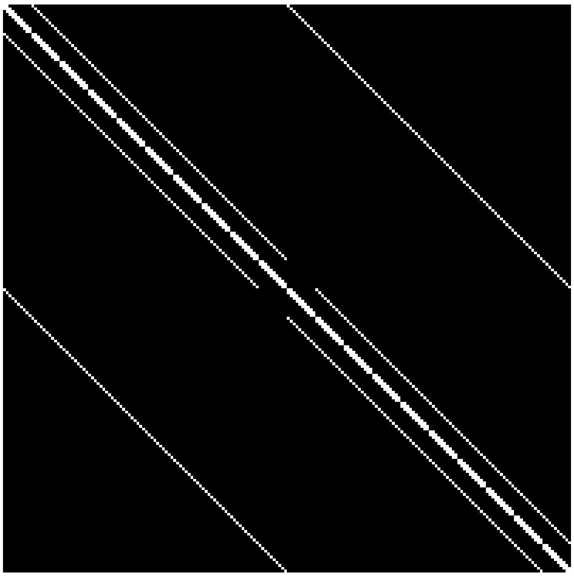


Figure 2.3: Visualization of Matrix $\mathbf{A}$. Black pixels denote zero entries of the matrix, white pixels potentially non-zero entries.

vector of size $M \cdot N$. If these vectors are stacked for all (i, j), the completely unwrapped stencil matrix $\langle \mathcal{S} \rangle$ results, which has the structure

$$\langle \mathcal{S} \rangle = \begin{pmatrix} \langle \mathcal{S}_{0,0} \rangle \\ \vdots \\ \langle \mathcal{S}_{N,M} \rangle \end{pmatrix} . \tag{2.37}$$

Matrix $\mathbf{A}$ is very large and sparse. Figure 2.3 visualizes the non-zero entries of $\mathbf{A}$ for an image of size 10×10. In general, the number of non-zero entries of $\mathbf{A}$ for an image of size $N \times M$ can be calculated as $12 \cdot MN - 4(M + N)$. Thus, it is linear w.r.t. the total number of pixels, $M \cdot N$. The number of non-zero entries can be determined by the sum of the following parts:

- diagonal: $2 \cdot M \cdot N$
- second diagonal: $2 \cdot M \cdot N$
- number of central pixels $\cdot 4 \cdot 2 = (M - 2) \cdot (N - 2) \cdot 4 \cdot 2$
- number of border pixels $\cdot 3 \cdot 2 = (2 \cdot (M - 2) + 2 \cdot (N - 2)) \cdot 3 \cdot 2$
- number of corner pixels $\cdot 2 \cdot 2 = 4 \cdot 3 \cdot 2$

Together with the total matrix size of $(2MN)^2$, the matrix density can be calculated. It is defined as the percentage of non-zero entries in a matrix and is given by

$$\rho(\mathbf{A}) = \frac{12MN - 4(M + N)}{(2MN)^2} . \tag{2.38}$$

This evaluates to very low values: For an image size of 100×100 pixels, the density of $\mathbf{A}$ is $0.0298\ \%$ and for an image size of 1242×375, as in the KITTI 15 dataset, the density is $0.000643\ \%$. The special properties of this matrix are used in the following to find the solution of the corresponding linear matrix equation.

2.4 Iterative Solvers

In order to solve linear systems of equations of large sizes, Gauss elimination is not feasible. But, since the matrices are highly sparse, iterative solvers can be used to find a solution to the matrix equation. Iterative solvers decompose the matrix $\mathbf{A}$ into a sum of two matrices $\mathbf{A}_1$ and $\mathbf{A}_2$, where $\mathbf{A}_1$ should be a reasonable approximation of $\mathbf{A}$ and easily invertible. Then from

$$(\mathbf{A}_1 + \mathbf{A}_2) \cdot \mathbf{x} = \mathbf{b} \Leftrightarrow \mathbf{A}_1\mathbf{x} = \mathbf{b} - \mathbf{A}_2\mathbf{x} \ , \tag{2.39}$$

a fixed-point iteration can be introduced via

$$\mathbf{A}_1\mathbf{x}^{k+1} = \mathbf{b} - \mathbf{A}_2\mathbf{x}^k \Leftrightarrow \mathbf{x}^{k+1} = \mathbf{A}_1^{-1}(\mathbf{b} - \mathbf{A}_2\mathbf{x}^k) \ . \tag{2.40}$$

Jacobi Method The Jacobi method uses a decomposition of matrix $\mathbf{A}$ into the diagonal on the one hand, and the strictly lower and upper triangular part of $\mathbf{A}$ on the other hand. The resulting iteration step then reads for all elements i of $\mathbf{x}$ [Saa03]

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^k \right) \ . \tag{2.41}$$

When using this method to solve Equations (2.27) and (2.28), the resulting Jacobi step for the method of Horn and Schunck becomes

$$u_{ij}^{k+1} = \frac{-[\mathbf{J}_{13}]_{ij} - \left([\mathbf{J}_{12}]_{ij} \cdot v_{ij}^k - \alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{u_{\tilde{i},\tilde{j}}^k}{h_l^2} \right)}{[\mathbf{J}_{11}]_{ij} + \alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}} \ , \tag{2.42}$$

$$v_{ij}^{k+1} = \frac{-[\mathbf{J}_{23}]_{ij} - \left([\mathbf{J}_{12}]_{ij} \cdot u_{ij}^k - \alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{v_{\tilde{i},\tilde{j}}^k}{h_l^2} \right)}{[\mathbf{J}_{22}]_{ij} + \alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}} \ . \tag{2.43}$$

As Saad [Saa03] proves, the Jacobi method converges to the desired solution, if $\mathbf{A}$ is strictly diagonally dominant or an irreducibly diagonally dominant matrix.

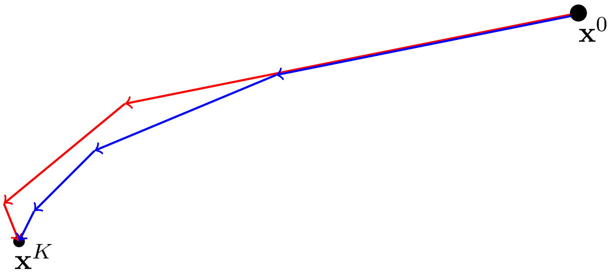


Figure 2.4: Schematic comparison of the Gauss-Seidel method (blue) and its SOR variant (red) in 2D. By using larger step sizes, SOR converges faster from the starting point $\mathbf{x}^0$ to the solution $\mathbf{x}^K$.

Gauss-Seidel Method In comparison to the Jacobi method, the Gauss-Seidel method uses a different matrix decomposition. It uses the diagonal and strictly lower triangle part as $\mathbf{A}_1$ and the strictly upper triangle part as $\mathbf{A}_2$. This yields the general iteration step [Saa03]

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij} x_j^{k+1} - \sum_{j>i} a_{ij} x_j^k \right) . \tag{2.44}$$

For the method of Horn and Schunck, the iteration step reads

$$u_{ij}^{k+1} = \frac{-[\mathbf{J}_{13}]_{ij} - \left([\mathbf{J}_{12}]_{ij} \cdot v_{ij}^k - \alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l^-(i,j)} \frac{u_{\tilde{i},\tilde{j}}^{k+1}}{h_l^2} - \alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l^+(i,j)} \frac{u_{\tilde{i},\tilde{j}}^{k}}{h_l^2} \right)}{[\mathbf{J}_{11}]_{ij} + \alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}} , \tag{2.45}$$

$$v_{ij}^{k+1} = \frac{-[\mathbf{J}_{23}]_{ij} - \left([\mathbf{J}_{12}]_{ij} \cdot u_{ij}^{k+1} - \alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l^-(i,j)} \frac{v_{\tilde{i},\tilde{j}}^{k+1}}{h_l^2} - \alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l^+(i,j)} \frac{v_{\tilde{i},\tilde{j}}^{k}}{h_l^2} \right)}{[\mathbf{J}_{22}]_{ij} + \alpha \sum\limits_{l \in x,y} \sum\limits_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}} . \tag{2.46}$$

Convergence of the Gauss-Seidel method is given, if $\mathbf{A}$ is symmetric and positive definite [Saa03], which is the case for the matrix resulting from the method of Horn and Schunck.

Successive Over-Relaxation (SOR) Successive Over-Relaxation (SOR) is an extension to the Gauss-Seidel method developed by Young [You71]. It has the goal of speeding up the solution of the linear system, i.e. reducing the number of iterations to reach convergence. Figure 2.4 schematically visualizes the idea of SOR for a 2D scenario with an initial estimate $\mathbf{x}^0$ and the solution $\mathbf{x}^K$. Instead of using the Gauss-Seidel iteration steps (blue), SOR (red) uses a linear combination of the old point $\mathbf{x}^k$ and the result of the Gauss-Seidel algorithm $\mathbf{x}^{\mathrm{GS}}$ and balances them with a parameter $\omega \in (0, 2)$:

$$x_i^{k+1} = (1 - \omega) \cdot x_i^k + \omega \cdot x_i^{\mathrm{GS}} . \tag{2.47}$$

For $\omega \in (0,1)$, the result interpolates between the two points and for a value larger than 1 (as shown in Figure 2.4), SOR extrapolates the result of the Gauss-Seidel method. This has the idea of a faster convergence to the solution. For $\omega = 1$, this method collapses to the Gauss-Seidel method. Since SOR also converges for symmetric positive definite matrices [Saa03] and due to its faster convergence it is the favorable choice for the solution of linear systems of equations resulting from variational optical flow methods.

2.5 Advanced Constancy Assumptions

In the following section, two extensions to the method of Horn and Schunck will be presented. First, in addition to the brightness constancy assumption as described in Section 2.3.1, an assumption on the image gradient is formulated. This allows to incorporate additional image information which is more independent of brightness changes. Second, a method is presented to omit the linearization of the data term assumptions, which allows for a higher precision of the solution.

2.5.1 Gradient Constancy Assumption

The brightness constancy assumption (see Equation (2.9)) is central to the method of Horn and Schunck. To improve the robustness, an additional assumption on the gradient can be posed, the gradient constancy assumption (GCA). Its advantage is that it is robust against global additive brightness changes between the two images, which would contradict the brightness constancy assumption. It is given as

$$\nabla f(\mathbf{x}, t) = \nabla f(\mathbf{x} + \mathbf{u}, t+1) \ . \tag{2.48}$$

Analogously to Equation (2.10), the gradient constancy assumption can be approximated by a linearized version

$$f_{xx}u + f_{xy}v + f_{xt} \approx 0 \ , \tag{2.49}$$

$$f_{yx}u + f_{yy}v + f_{yt} \approx 0 \ . \tag{2.50}$$

When integrating the assumptions in the energy functional, a motion tensor notation, analogous to Equation (2.13) can be introduced:

$$\mathbf{J}_{\text{gca}} = \mathbf{J}_{\text{gca},x} + \mathbf{J}_{\text{gca},y} \ , \tag{2.51}$$

$$\tag{2.52}$$

with

$$\mathbf{J}_{\text{gca},x} = \begin{pmatrix} f_{xx} \\ f_{xy} \\ f_{xt} \end{pmatrix} \cdot \begin{pmatrix} f_{xx} \\ f_{xy} \\ f_{xt} \end{pmatrix}^\top \ , \tag{2.53}$$

$$\mathbf{J}_{\text{gca},y} = \begin{pmatrix} f_{yx} \\ f_{yy} \\ f_{yt} \end{pmatrix} \cdot \begin{pmatrix} f_{yx} \\ f_{yy} \\ f_{yt} \end{pmatrix}^\top \ . \tag{2.54}$$

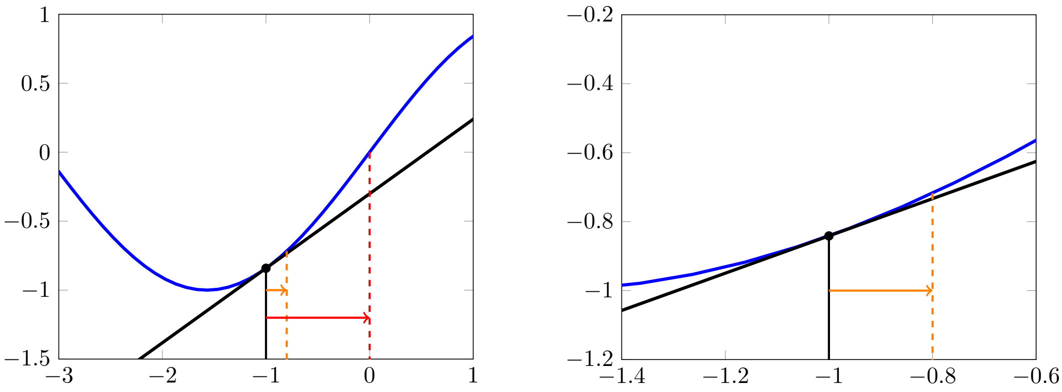


Figure 2.5: Limitations of Taylor approximations. First-order Taylor expansion (black) for two arbitrary functions (blue). For small distances around the expansion point the linear function is a good approximation (orange), in contrast to larger distances (red). For a smooth function the Taylor approximation holds for larger distances.

The goal is to use a combination of both assumptions in the energy functional. For this, the brightness constancy assumption motion tensor introduced in Equation (2.13) is renamed to $\mathbf{J}_{\text{bca}}$ and again the extended flow abbreviation $\mathbf{w} = (\mathbf{u}^\top, 1) = (u, v, 1)^\top$ is used. Then the new energy functional integrating both assumptions reads

$$E(\mathbf{u}) = \int_\Omega \left(\mathbf{w}^\top \mathbf{J}_{\text{bca}} \mathbf{w} + \gamma \cdot \mathbf{w}^\top \mathbf{J}_{\text{gca}} \mathbf{w} + \alpha \cdot \left(|\nabla u|^2 + |\nabla v|^2 \right) \right) d\mathbf{x} \; . \tag{2.55}$$

The newly introduced parameter γ balances the two assumptions. Analogously to Equations (2.18) and (2.19), the Euler-Lagrange equations for this new energy functional are extended to

$$\mathbf{J}_{\text{bca},11} \cdot u + \mathbf{J}_{\text{bca},12} \cdot v + \mathbf{J}_{\text{bca},13} + \gamma \cdot \left(\mathbf{J}_{\text{gca},11} \cdot u + \mathbf{J}_{\text{gca},12} \cdot v + \mathbf{J}_{\text{gca},13} \right) - \alpha \, \Delta u = 0 \; , \tag{2.56}$$

$$\mathbf{J}_{\text{bca},21} \cdot u + \mathbf{J}_{\text{bca},22} \cdot v + \mathbf{J}_{\text{bca},23} + \gamma \cdot \left(\mathbf{J}_{\text{gca},21} \cdot u + \mathbf{J}_{\text{gca},22} \cdot v + \mathbf{J}_{\text{gca},23} \right) - \alpha \, \Delta v = 0 \; . \tag{2.57}$$

Again, the boundary conditions are $\mathbf{n}^\top \nabla u = 0$ and $\mathbf{n}^\top \nabla v = 0$.

2.5.2 Constancy Assumption without Linearization

The linearization of the optical flow constraint as presented in Equation (2.10) implicitly assumes the flow to be small or the image to be very smooth. Figure 2.5 visualizes the problems that may come with linearization.

In order to overcome these problems, the energy functional can be formulated without the linearization step. Subsequently, it reads for the brightness constancy assumption

$$E(\mathbf{u}) = \int_\Omega \left(f(\mathbf{x} + \mathbf{u}, t + 1) - f(\mathbf{x}, t) \right)^2 + \alpha \cdot \left(|\nabla u|^2 + |\nabla v|^2 \right) d\mathbf{x} \; . \tag{2.58}$$

In comparison to the previous approaches, the resulting energy functional is now non-convex with possibly multiple local minima.

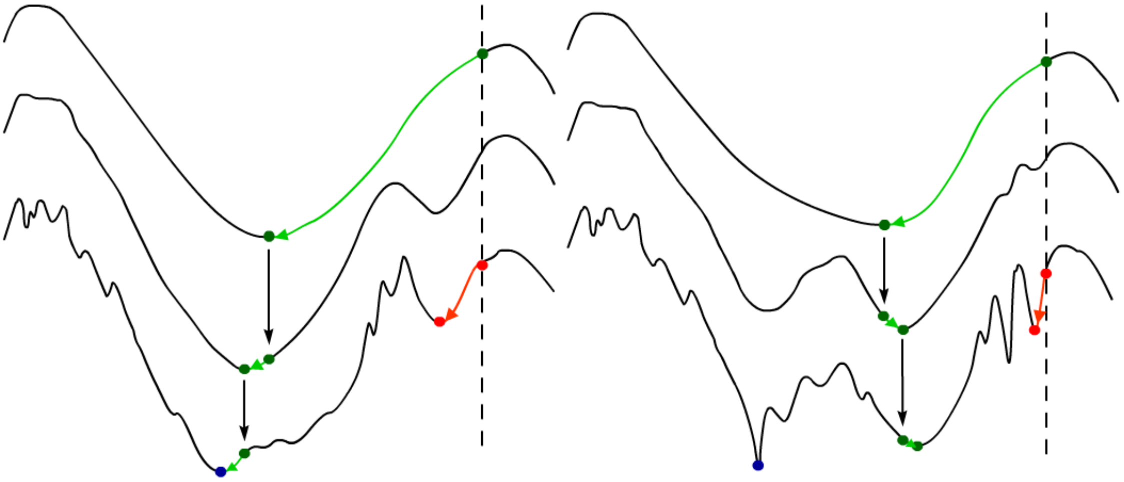


Figure 2.6: Red: Non-hierarchical minimization. Green: Coarse-to-fine minimization. Blue: Global minimum. *Left:* Global optimum found. *Right:* Useful local minimum found. Image Source: [Bru06].

To solve an energy functional like this, the warping approach was introduced by Brox *et al.* [BBPW04]. It makes use of a coarse-to-fine scheme to solve a non-convex function. Figure 2.6 illustrates the idea: Instead of directly minimizing the energy functional using the original image (red), where the functional might contain multiple local minima, the method makes use of an image pyramid. An image pyramid is a sequence of the same image where the resolution is iteratively reduced. The coarse-to-fine minimization first searches for an optimum using a low-resolution version of the images. Then, this intermediate solution is used as an initialization for the next level of higher resolution, which is repeated until the original image resolution is reached. This coarse-to-fine minimization (green) has the goal to either converge to the global minimum (blue) as depicted on the left or to a good local minimum as shown on the right.

In the following, the minimization using the warping approach will be derived. First, the necessary conditions of the functional in Equation (2.58) have to be derived. The Euler-Lagrange equations read

$$f_x(\mathbf{x}+\mathbf{u},t+1)\cdot(f(\mathbf{x}+\mathbf{u},t+1)-f(\mathbf{x},t))-\alpha\,\Delta u=0\;, \tag{2.59}$$

$$f_y(\mathbf{x}+\mathbf{u},t+1)\cdot(f(\mathbf{x}+\mathbf{u},t+1)-f(\mathbf{x},t))-\alpha\,\Delta v=0\;, \tag{2.60}$$

with boundary conditions $\mathbf{n}^\top\nabla u=0$ and $\mathbf{n}^\top\nabla v=0$.

Second, the image pyramid is introduced. For each level k, the image data is downsampled to the corresponding resolution. The resolution is determined by the coarsening factor η, which controls the reduction of resolution per level. The two downsampled images are then denoted by $f^k(\mathbf{x},t)$ and $f^k(\mathbf{x},t+1)$. Similarly, the flow on each level is given by $\mathbf{u}^k$. Since the method uses the solution of the previous resolution level as an initialization (see Figure 2.6), only the difference flow $\mathbf{du}^k$ must be determined. It is defined by the splitting

$$\mathbf{u}^{k+1}=\mathbf{u}^k+\mathbf{du}^k=(u^k,v^k)^\top+(du^k,dv^k)^\top\;, \tag{2.61}$$

where $\mathbf{u}^k$ is the flow on the coarser level which is upsampled to the next finer resolution level using bilinear interpolation.

This fixed-point iteration can then be introduced into Equations (2.59) and (2.60), where the smoothness term is implicit and the data term is semi-implicit:

$$f_x^k(\mathbf{x}+\mathbf{u}^k,t+1)\cdot\Big(f^k(\mathbf{x}+\mathbf{u}^{k+1},t+1)-f^k(\mathbf{x},t)\Big)-\alpha\,\Delta u^{k+1}=0\ , \tag{2.62}$$

$$f_y^k(\mathbf{x}+\mathbf{u}^k,t+1)\cdot\Big(f^k(\mathbf{x}+\mathbf{u}^{k+1},t+1)-f^k(\mathbf{x},t)\Big)-\alpha\,\Delta v^{k+1}=0\ . \tag{2.63}$$

Using the splitting defined in Equation (2.61), the term $f^k(\mathbf{x}+\mathbf{u}^{k+1},t+1)=f^k(\mathbf{x}+\mathbf{u}^k+\mathbf{du}^k,t+1)$ can be linearized w.r.t. $\mathbf{du}^k$:

$$f^k(\mathbf{x}+\mathbf{u}^{k+1},t+1)=f^k(\mathbf{x}+\mathbf{u}^k+\mathbf{du}^k,t+1) \tag{2.64}$$

$$\approx f^k(\mathbf{x}+\mathbf{u}^k,t+1)+f_x^k(\mathbf{x}+\mathbf{u}^k,t+1)\cdot du^k + f_y^k(\mathbf{x}+\mathbf{u}^k,t+1)\cdot dv^k\ . \tag{2.65}$$

Finally, the first Euler-Lagrange equation becomes

$$\begin{aligned} f_x^k(\mathbf{x}+\mathbf{u}^k,t+1)\cdot\Big(&f_x^k(\mathbf{x}+\mathbf{u}^k,t+1)\cdot du^k\\ &+f_y^k(\mathbf{x}+\mathbf{u}^k,t+1)\cdot dv^k\\ &+f^k(\mathbf{x}+\mathbf{u}^k,t+1)-f^k(\mathbf{x},t)\Big)-\alpha\,\Delta(u^k+du^k)=0 \end{aligned} \tag{2.66}$$

$$\Leftrightarrow \mathbf{J}_{11}^k\cdot du^k+\mathbf{J}_{12}^k\cdot dv^k+\mathbf{J}_{13}^k-\alpha\,\Delta(u^k+du^k)=0\ , \tag{2.67}$$

where the following definition of the motion tensor $\mathbf{J}^k$ is used

$$\mathbf{J}^k=\begin{pmatrix} f_x^k(\mathbf{x}+\mathbf{u}^k,t+1)\\ f_y^k(\mathbf{x}+\mathbf{u}^k,t+1)\\ f^k(\mathbf{x}+\mathbf{u}^k,t+1)-f^k(\mathbf{x},t)\end{pmatrix}\cdot\begin{pmatrix} f_x^k(\mathbf{x}+\mathbf{u}^k,t+1)\\ f_y^k(\mathbf{x}+\mathbf{u}^k,t+1)\\ f^k(\mathbf{x}+\mathbf{u}^k,t+1)-f^k(\mathbf{x},t)\end{pmatrix}^\top\ . \tag{2.68}$$

Analogously, the second Euler-Lagrange equation reads

$$\mathbf{J}_{21}^k\cdot du^k+\mathbf{J}_{22}^k\cdot dv^k+\mathbf{J}_{23}^k-\alpha\,\Delta(v^k+dv^k)=0\ . \tag{2.69}$$

In order to compute the motion tensor entries, $f^k(\mathbf{x}+\mathbf{u}^k,t+1)$ has to be determined. It is based on the second image on resolution level k, given by $f^k(\mathbf{x},t+1)$ which is warped by the already computed motion $\mathbf{u}^k$. This is done using backward registration. The resulting backward registered image $\tilde{f}^k$ has then for every position $\mathbf{x}$ the value $\tilde{f}^k(\mathbf{x})=f^k(\mathbf{x}+\mathbf{u}^k,t+1)$. For a good flow estimation, $\tilde{f}^k(\mathbf{x})$ strongly resembles the first image, $f^k(\mathbf{x},t)$.

Figure 2.7 visualizes the backward registration: To compute the red marked pixel of the backward registered image on the right, the corresponding flow vector at this location is used. Its displacement possibly points to a non-grid location in the second image (green). To then determine the resulting intensity value, the 4 neighboring pixel values (blue) are used to calculate the value with bilinear interpolation.

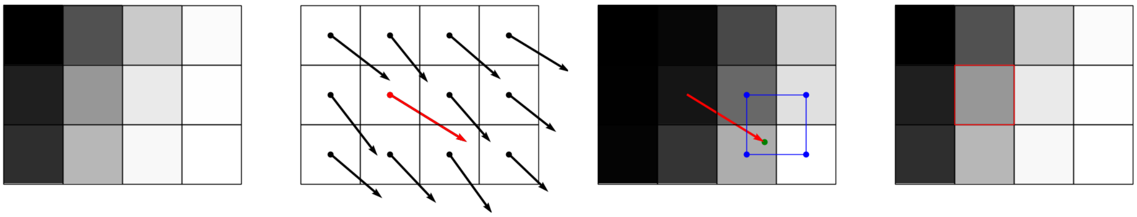


Figure 2.7: Backward registration. *From left to right:* First image $f(\mathbf{x},t)$, estimated flow field $\mathbf{u}(\mathbf{x})$, second image $f(\mathbf{x},t+1)$, backward registered result $\tilde{f}(\mathbf{x})$. In order to determine the value of the red surrounded pixel, the corresponding flow (red) determines the four values used for bilinear interpolation (blue).

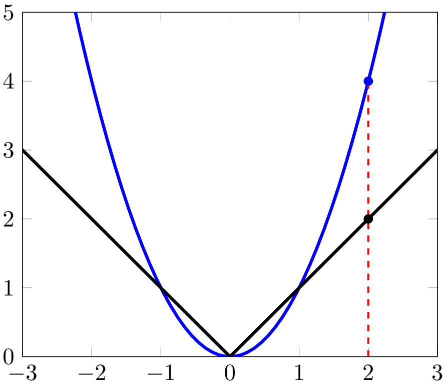


Figure 2.8: L_1 (black) and L_2 norm (blue). Outliers (red) contribute with quadratic weight.

2.6 Sub-quadratic Penalizers

The models for variational optical flow estimation that were presented so far can be improved regarding their robustness with respect to outliers. Up to now, the data term and the smoothness term are squared, which leads to a well-posed problem formulation and a linear system of equations. However, Figure 2.8 shows a problem with this quadratic penalizer: For an outlier (red) of either the smoothness or the constancy assumptions, the quadratic penalizer assigns a very high energy. Since the energy is integrated over the whole image domain, this leads to a large influence of outliers compared to inliers. If a sub-quadratic penalizer is used, this effect can be reduced, and the influence of outliers can be attenuated.

Outliers sometimes occur naturally. An example for the data term would be an object that gets occluded from one frame to the other, which would be an outlier for the constancy assumptions. For the smoothness term, an example would be two neighboring objects of different motion, at their boundary the flow field will show a discontinuity. This violates the smoothness assumption and would thus lead to a high error contribution in the smoothness term when using a quadratic penalizer. The solution for this is creating a sub-quadratic energy term, which makes the optical flow estimation more robust against outliers, which is why it is also referred to as *robustification*. In order to achieve this, a non-linear penalizer function $\Psi(s^2)$ is introduced, which is applied to the squared term in order to make it sub-quadratic.

In the following, first two robustification approaches will be explained: sub-quadratic data terms and sub-quadratic smoothness terms. Then, an overview of frequently used penalizer functions Ψ is given.

2.6.1 Sub-quadratic Data Terms

Any of the data terms presented so far can be combined with a robust penalizer function Ψ. When using it on the energy functional of the method of Horn and Schunck as in Equation (2.14), the new functional reads

$$E(\mathbf{u}) = \int_\Omega \Big(\Psi(\mathbf{w}^\top \mathbf{J} \mathbf{w}) + \alpha \cdot \big(|\nabla u|^2 + |\nabla v|^2\big) \Big) \, d\mathbf{x} \; , \tag{2.70}$$

where again the dual flow definition $\mathbf{w} = (\mathbf{u}^\top, 1) = (u, v, 1)^\top$ is used. For this functional, the necessary conditions read:

$$\Psi'(\mathbf{w}^\top \mathbf{J} \mathbf{w}) \, (\mathbf{J}_{11} \cdot u + \mathbf{J}_{12} \cdot v + \mathbf{J}_{13}) - \alpha \, \Delta u = 0 \; , \tag{2.71}$$

$$\Psi'(\mathbf{w}^\top \mathbf{J} \mathbf{w}) \, (\mathbf{J}_{21} \cdot u + \mathbf{J}_{22} \cdot v + \mathbf{J}_{23}) - \alpha \, \Delta v = 0 \; , \tag{2.72}$$

with boundary conditions $\mathbf{n}^\top \nabla u = 0$ and $\mathbf{n}^\top \nabla v = 0$. Note that the only difference to the previous Euler-Lagrange equations given in Equations (2.18) and (2.19) is the multiplication of Ψ'. The function Ψ is chosen in a way that if the argument of Ψ' is large – which corresponds to a potential outlier – this leads to a down-weighting of the constancy assumption in the Euler-Lagrange equations.

Compared to the previous Euler-Lagrange equations, these equations are now non-linear. After discretization, a non-linear system of equations is obtained:

$$0 = [\Psi']_{ij} \left([\mathbf{J}_{11}]_{ij} \cdot u_{ij} + [\mathbf{J}_{12}]_{ij} \cdot v_{ij} + [\mathbf{J}_{13}]_{ij} \right) - \alpha \sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \sum \mathcal{S}_{i,j,\tilde{i},\tilde{j}} \cdot u_{i,j} \; , \tag{2.73}$$

$$0 = [\Psi']_{ij} \left([\mathbf{J}_{21}]_{ij} \cdot u_{ij} + [\mathbf{J}_{22}]_{ij} \cdot v_{ij} + [\mathbf{J}_{23}]_{ij} \right) - \alpha \sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \sum \mathcal{S}_{i,j,\tilde{i},\tilde{j}} \cdot v_{i,j} \; , \tag{2.74}$$

with $[\Psi']_{ij} = \Psi' \left([\mathbf{w}]_{ij}^\top [\mathbf{J}]_{ij} [\mathbf{w}]_{ij} \right)$.

To solve such a non-linear system of equations, the *lagged non-linearity* or *Kačanov* method [KNPS68] is used. It makes use of a second fixed-point iteration, where the non-linear parts of the equation are evaluated with u and v taken from the old iteration step l, and the rest from the new iteration step $l + 1$:

$$0 = [\Psi']_{ij}^l \left([\mathbf{J}_{11}]_{ij} \cdot u_{ij}^{l+1} + [\mathbf{J}_{12}]_{ij} \cdot v_{ij}^{l+1} + [\mathbf{J}_{13}]_{ij} \right) - \alpha \sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \sum \mathcal{S}_{i,j,\tilde{i},\tilde{j}} \cdot u_{i,j}^{l+1} \; , \tag{2.75}$$

$$0 = [\Psi']_{ij}^l \left([\mathbf{J}_{21}]_{ij} \cdot u_{ij}^{l+1} + [\mathbf{J}_{22}]_{ij} \cdot v_{ij}^{l+1} + [\mathbf{J}_{23}]_{ij} \right) - \alpha \sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \sum \mathcal{S}_{i,j,\tilde{i},\tilde{j}} \cdot v_{i,j}^{l+1} \; . \tag{2.76}$$

After evaluating the non-linear parts, they are kept fixed and a linear system of equations remains, which can then be solved e.g. by the SOR method. Thus, the lagged non-linearity method transforms a non-linear system of equations into a series of linear systems of equations. Note that the lagged non-linearity fixed-point iteration, also denoted as outer iteration, is different from the solvers mentioned in Section 2.3.4, which refer to the inner iteration. Listing 2.1 gives a schematic overview on the two nested iterations.

Listing 2.1 Schematic overview of optical flow calculation with non-linear terms.

```
for l := 0 to num_outer_iterations {
        psi = compute_nonlinearities(u,v);
        for k := 0 to num_inner_iterations {
                u,v = SOR_iteration(u,v,psi);
        }
}
```

2.6.2 Sub-quadratic Smoothness Terms

Similar to data terms, the smoothness term can be robustified. Accordingly, using the flow abbreviation $\mathbf{w} = (\mathbf{u}^\top, 1) = (u, v, 1)^\top$, the energy functional reads

$$E(\mathbf{u}) = \int_\Omega \Big(\mathbf{w}^\top \mathbf{J} \mathbf{w} + \alpha \cdot \Psi\big(|\nabla u|^2 + |\nabla v|^2\big) \Big) \; d\mathbf{x} \; . \tag{2.77}$$

The Euler Lagrange equations are then given by

$$\mathbf{J}_{11} \cdot u + \mathbf{J}_{12} \cdot v + \mathbf{J}_{13} - \alpha \; \mathrm{div} \left(\Psi'\big(|\nabla u|^2 + |\nabla v|^2\big) \; \nabla u \right) = 0 \; , \tag{2.78}$$

$$\mathbf{J}_{21} \cdot u + \mathbf{J}_{22} \cdot v + \mathbf{J}_{23} - \alpha \; \mathrm{div} \left(\Psi'\big(|\nabla u|^2 + |\nabla v|^2\big) \; \nabla v \right) = 0 \; , \tag{2.79}$$

with the boundary conditions $\mathbf{n}^\top \cdot \Psi'\big(|\nabla u|^2 + |\nabla v|^2\big) \cdot \nabla u = 0$ and $\mathbf{n}^\top \cdot \Psi'\big(|\nabla u|^2 + |\nabla v|^2\big) \cdot \nabla v = 0$. As before, the Euler-Lagrange equations are non-linear in u and v. After discretization, a system of non-linear equations is obtained:

$$0 = [\mathbf{J}_{11}]_{ij} \cdot u_{ij} + [\mathbf{J}_{12}]_{ij} \cdot v_{ij} + [\mathbf{J}_{13}]_{ij} - \alpha \sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \sum \mathcal{S}_{i,j,\tilde{i},\tilde{j}} \cdot u_{i,j} \; , \tag{2.80}$$

$$0 = [\mathbf{J}_{21}]_{ij} \cdot u_{ij} + [\mathbf{J}_{22}]_{ij} \cdot v_{ij} + [\mathbf{J}_{23}]_{ij} - \alpha \sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \sum \mathcal{S}_{i,j,\tilde{i},\tilde{j}} \cdot v_{i,j} \; . \tag{2.81}$$

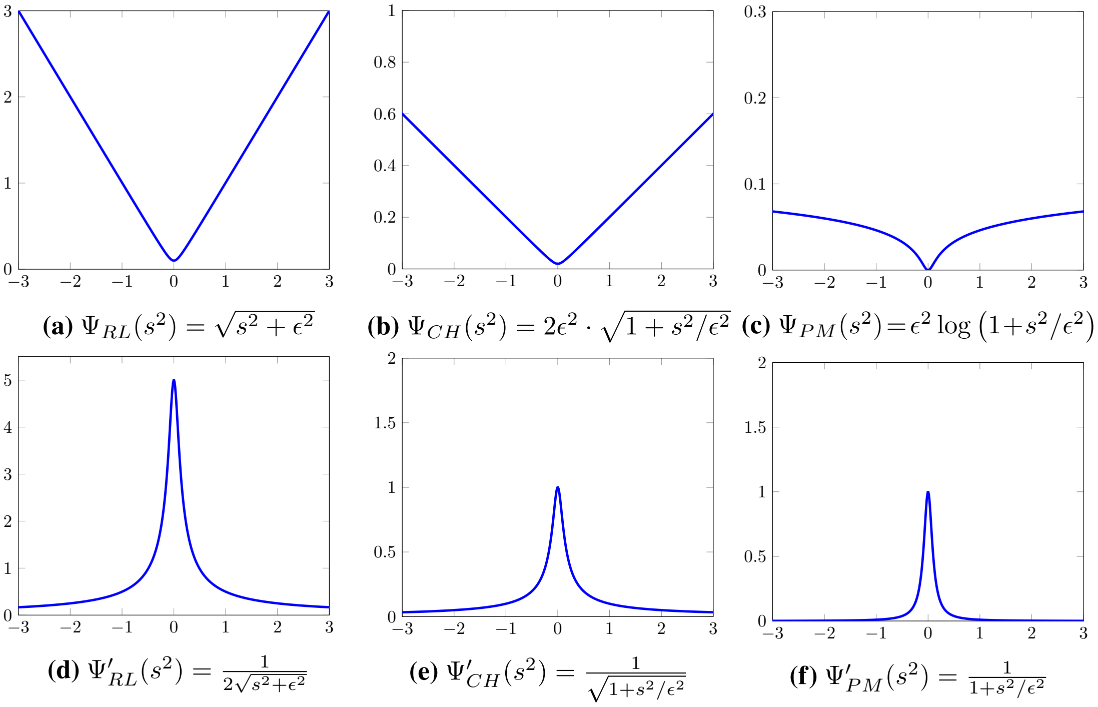
Again, a spatially variant stencil is used, which when omitting the boundary conditions reads

$$\mathcal{S}_{i,j} = \begin{array}{|c|c|c|} \hline & \frac{[\Psi']_{i,j+1} + [\Psi']_{ij}}{2h_y^2} & \\ \hline \frac{[\Psi']_{i-1,j} + [\Psi']_{ij}}{2h_x^2} & \begin{array}{c} -\frac{[\Psi']_{i-1,j} + 2[\Psi']_{ij} + [\Psi']_{i+1,j}}{2h_x^2} \\ -\frac{[\Psi']_{i,j-1} + 2[\Psi']_{ij} + [\Psi']_{i,j+1}}{2h_y^2} \end{array} & \frac{[\Psi']_{i+1,j} + [\Psi']_{ij}}{2h_x^2} \\ \hline & \frac{[\Psi']_{i,j-1} + [\Psi']_{ij}}{2h_y^2} & \\ \hline \end{array} \; , \tag{2.82}$$

where $[\Psi']_{i,j} = \Psi'\Big([\nabla u]_{ij}^2 + [\nabla v]_{ij}^2\Big)$. Note that in contrast to before, the stencil values are non-linear in u and v due to the Ψ' function. In order to solve the resulting non-linear system of equations, again the lagged non-linearity method is used. For this, an additional outer iteration updates only the non-linearities while they are kept fixed during the inner iteration.

2.6.3 Penalizer Functions

In the following, three commonly used penalizer functions will be introduced. Additionally, they are plotted with their first derivative in Figure 2.9.



(a) $\Psi_{RL}(s^2) = \sqrt{s^2 + \epsilon^2}$ **(b)** $\Psi_{CH}(s^2) = 2\epsilon^2 \cdot \sqrt{1 + s^2/\epsilon^2}$ **(c)** $\Psi_{PM}(s^2) = \epsilon^2 \log\left(1 + s^2/\epsilon^2\right)$

(d) $\Psi'_{RL}(s^2) = \frac{1}{2\sqrt{s^2+\epsilon^2}}$ **(e)** $\Psi'_{CH}(s^2) = \frac{1}{\sqrt{1+s^2/\epsilon^2}}$ **(f)** $\Psi'_{PM}(s^2) = \frac{1}{1+s^2/\epsilon^2}$

Figure 2.9: Overview of penalizer functions. *From left to right:* Regularized Linear, Charbonnier, Perona-Malik. *Top row:* Penalizer function, *bottom row:* derivative. The regularization constant was set to $\epsilon = 0.1$.

Regularized Linear Penalizer The most obvious choice for a penalizer function that transforms a quadratic function to a linear function would be the square root. However, since the goal is to keep the functional differentiable, a regularized version with a small $\epsilon > 0$ is used. It is defined as

$$\Psi_{RL}(s^2) = \sqrt{s^2 + \epsilon^2}\ , \tag{2.83}$$

with the derivative given by

$$\Psi'_{RL}(s^2) = \frac{1}{2\sqrt{s^2 + \epsilon^2}}\ . \tag{2.84}$$

Charbonnier The penalizer proposed by Charbonnier *et al.* [CBAB97] is a slightly modified version of the regularized linear penalizer. It has the advantage that its derivative is bounded to 1, which is helpful when determining a stability criterion. Thus it is commonly preferred over the regularized linear penalizer. The formula is given as

$$\Psi_{CH}(s^2) = 2\epsilon^2 \cdot \sqrt{1 + \frac{s^2}{\epsilon^2}}\ , \tag{2.85}$$

with the derivative

$$\Psi'_{CH}(s^2) = \frac{1}{\sqrt{1 + \frac{s^2}{\epsilon^2}}}\ . \tag{2.86}$$

Perona-Malik Finally, the penalizer by Perona and Malik [PM90] is introduced:

$$\Psi_{PM}(s^2) = \epsilon^2 \log\left(1 + \frac{s^2}{\epsilon^2}\right) , \tag{2.87}$$

with its derivative

$$\Psi'_{PM}(s^2) = \frac{1}{1 + \frac{s^2}{\epsilon^2}} . \tag{2.88}$$

As Maurer describes, the Perona-Malik penalizer is not only sub-quadratic as all penalizers so far, it is also sublinear, which makes it also non-convex [Mau19].

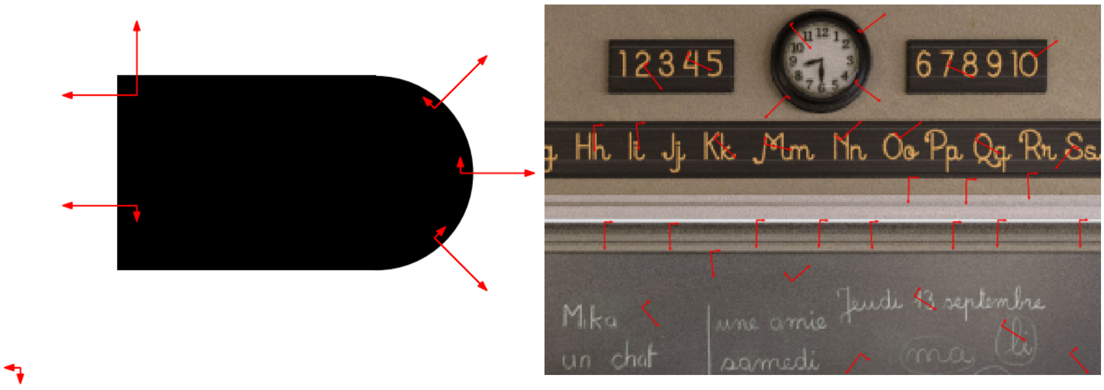
2.7 Advanced Smoothness Terms

So far, the regularization approaches are limited in two ways. First, the regularization terms up to now lead to an isotropic smoothing behavior, where isotropic means that for every pixel location, the smoothing is the same in any local direction. However, a difference between smoothing behaviors e.g. along and across image edges can be helpful. Second, so far only a first-order regularization was used, which only penalizes the first derivative of the flow, yet this can be extended to higher orders.

2.7.1 Anisotropic Regularization

In order to make the extension from an isotropic regularization to a directional regularization, first a look into the corresponding diffusion processes has to be done. Afterwards the regularization tensor is presented, which is used to derive local directions. Afterwards, the integration into the energy functional is discussed.

Anisotropic Diffusion There exists a strong relation between smoothness terms and diffusion processes. While the smoothness term models a smooth motion field as a desired state, the corresponding Euler-Lagrange equations as the necessary conditions models the smoothing process, i.e. the diffusion. This diffusion process is given in the Euler-Lagrange equations as a divergence expression applied to the gradient of the field. In the Euler-Lagrange equations of the method by Horn and Schunk as given in Equations (2.15) and (2.16), the Laplacian $\Delta u = \operatorname{div}(\nabla u)$ represents homogeneous linear diffusion. When introducing a robust function into the smoothness term, the resulting Euler-Lagrange equations as given in Equations (2.78) and (2.79) contain a scalar multiplier inside the divergence expression, $\operatorname{div}(\Psi' \cdot \nabla u)$, which represents non-linear isotropic diffusion. In order to further extend the diffusion process, instead of a scalar multiplier, a matrix multiplier can be used inside the divergence expression: $\operatorname{div}(\mathbf{D} \cdot \nabla u)$. When selecting a matrix with two different eigenvalues, their intensity controls the smoothing behavior individually in the direction of the corresponding eigenvectors. This allows for an anisotropic smoothing behavior that is e.g. adaptive to local image structures and allows for a different smoothing along and across edges. In order to model the diffusion tensor, two fitting directions need to be determined.



(a) Schematic visualization for a test image.

(b) Computed regularization tensors entries for sparse positions selected by non-maxima suppression.

Figure 2.10: Visualization of the regularization tensor. For selected positions, both eigenvectors scaled by their eigenvalues are depicted.

Regularization Tensor Essential to the anisotropic diffusion are two local directions that control the smoothing behavior. In order to model anisotropic regularization using anisotropic diffusion processes, they need to be determined. For this, directions derived from the local image structure are useful. To this end, the regularization tensor can be computed, which is defined per pixel. It was introduced as a *complementary regularizer* by Zimmer *et al.* [ZBW11] and is defined as follows:

$$\mathbf{R}_\rho = K_\rho * \left[\begin{pmatrix} f_x \\ f_y \end{pmatrix} \cdot \begin{pmatrix} f_x \\ f_y \end{pmatrix}^\top + \gamma \cdot \left(\begin{pmatrix} f_{xx} \\ f_{xy} \end{pmatrix} \cdot \begin{pmatrix} f_{xx} \\ f_{xy} \end{pmatrix}^\top + \begin{pmatrix} f_{yx} \\ f_{yy} \end{pmatrix} \cdot \begin{pmatrix} f_{yx} \\ f_{yy} \end{pmatrix}^\top \right) \right] , \tag{2.89}$$

where K_ρ is a Gaussian convolution kernel of size ρ. Note that the regularization tensor can be seen as a complement to the motion tensor as used in Equation (2.55) since it uses the same weight γ to balance the brightness and gradient derivative. However, it differs from the motion tensor by the fact that only the 2D spatial gradient is used and that it is additionally integrated over a neighborhood using the Gaussian convolution.

By then determining the eigenvectors and eigenvalues of this regularization tensor, information about the image structure can be obtained. Figure 2.10 visually explains the resulting vectors. Figure 2.10a shows a test image with schematic tensor visualizations: For selected image locations, the two eigenvectors scaled by their eigenvalues are depicted. At flat image locations with no edges, both eigenvalues are small, and the eigenvectors are arbitrary oriented. At image edges, the first eigenvalue is large with its eigenvector pointing across the edge, and at image corners, both eigenvalues are large. Figure 2.10b shows computed values for the depicted image which is a cropped image of the classroom sequence. Since the arrow visualization only allows to show sparse values, a non-maxima suppression was executed on the first eigenvalues and only the resulting positions are plotted. This leads to showing only distinctly selected non-overlapping positions where the first eigenvalue is a local maximum. The visualization shows that image structures are well recognized, note e.g. the border of the blackboard or the clock.

As shown, the eigenvectors of the regularization tensor provide a good way of encoding information about the image structure. In the following, they will be referred to as $\mathbf{r}_1$ and $\mathbf{r}_2$, with $\mathbf{r}_1$ pointing in the more dominant direction. Note that in contrast to the visualization in Figure 2.10, both of them are normalized to a length of 1, i.e. the corresponding eigenvalues are ignored.

Energy Functional Using the directional information obtained from the regularization tensor, the robust smoothness term as presented in Equation (2.77) can be extended. In order to do so, the squared gradient ∇u is replaced by a sum of the directional derivatives in direction of the regularization vectors $\mathbf{r}_1$ and $\mathbf{r}_2$, where the penalization is then applied to each direction individually. The directional derivatives $u_{\mathbf{r}_l}$ are computed as $\mathbf{r}_l^\top \nabla u$. Then the anisotropic robust regularization term as introduced by Zimmer *et al.* [ZBW11] reads

$$R(\mathbf{u}) = \int_\Omega \sum_{l=1}^{2} \Psi_l \Big((\mathbf{r}_l^\top \nabla u)^2 + (\mathbf{r}_l^\top \nabla v)^2 \Big) \, d\mathbf{x} \; . \tag{2.90}$$

Note that only the regularization term R is shown, for a full optical flow estimation, a suitable data term is needed. When deriving the corresponding Euler-Lagrange equations, they yield an anisotropic diffusion process, where the diffusion is steered by the directions $\mathbf{r}_1$ and $\mathbf{r}_2$.

2.7.2 Second-Order Regularization

The smoothness terms presented so far impose a penalty on the first derivative of the gradient, as in the original paper of Horn and Schunck. Also, as discussed in Section 2.6.2, a penalizer function helps with flow edges. This way constant – or due to the robustness piecewise constant – flow fields are favored. However, as for instance motivated by Figure 1.1, there are cases where a penalty on the second order makes sense, e.g. scenes with a distinct ego-motion. Forcing a small second derivative favors a flow field which is linear or piecewise linear.

In the following, two examples of a second-order regularization will be presented, first a direct penalization of the Hessian and afterwards an indirect coupling approach.

Hessian-based Regularization A straightforward way of penalizing the second derivative of the flow, is by making use of the Hessian $\mathcal{H}$. It is a matrix consisting of all second derivatives and is for a 2D case given by

$$\mathcal{H}u = \begin{pmatrix} u_{xx} & u_{xy} \\ u_{yx} & u_{yy} \end{pmatrix} \; . \tag{2.91}$$

This matrix is then used inside the second-order robust regularization term, which has been proposed by Demetz *et al.* [DSV+14] and reads

$$R(\mathbf{u}) = \int_\Omega \Psi \left(\|\mathcal{H}u\|_F^2 + \|\mathcal{H}v\|_F^2 \right) \, d\mathbf{x} \; , \tag{2.92}$$

where $\|\cdot\|_F^2$ is the Frobenius norm.

Coupling-based Regularization Instead of directly penalizing the second derivative of the flow, an indirect coupling approach can be used. It was first introduced by Bredies *et al.* [BKP10] and applied to optical flow by Braux-Zin *et al.* [BDB13]. The second-order coupling regularization term reads

$$R(\mathbf{u}) = \int_\Omega \inf_{\mathbf{a},\mathbf{b}} \left\{ \Psi\left(|\nabla u - \mathbf{a}|^2 + |\nabla v - \mathbf{b}|^2\right) + \beta \cdot \Psi\left(\|\mathcal{J}\mathbf{a}\|_F^2 + \|\mathcal{J}\mathbf{b}\|_F^2\right) \right\} d\mathbf{x} \,, \tag{2.93}$$

where $\mathbf{a}$ and $\mathbf{b}$ are two auxiliary fields and $\mathcal{J}$ is the Jacobian matrix. The structure of this regularization term needs further explanation. It consists of two parts, where the first part is a coupling term that enforces $\mathbf{a}$ to be similar to ∇u and $\mathbf{b}$ similar to ∇v. Thus, the auxiliary variables approximate the gradient of the flow field. The second term is a first-order penalization of the auxiliary variables. Since they already correspond to a first derivative, this can be regarded as a loosely-coupled second-order penalty on the flow. Both terms individually make use of a robust function Ψ. An additional parameter β balances the two terms against each other.

3 Order-Adaptive Regularization with an Isotropic Selection Scheme

In this chapter, the order-adaptive regularization approach for variational optical flow by Maurer *et al.* [MSB17b] will be presented. First, an introduction to advanced regularization terms will be given. Afterwards, the model of Maurer *et al.* [MSB17b] is presented, followed by its minimization. In the end, a detailed derivation of the necessary discretization is provided, which can then be solved using the methods presented before.

3.1 Method of Maurer *et al.*

The method of Maurer *et al.* [MSB17b] incorporates many of the ideas presented so far. Its general energy functional is given by

$$E(\mathbf{u}, c) = D(\mathbf{u}) + \alpha \cdot R(\mathbf{u}, c)\ . \tag{3.1}$$

As before, the functional is a weighted sum of a data term $D(\mathbf{u})$ and a regularization term $R(\mathbf{u}, c)$. New, however, is the order selection parameter c which appears in the regularization term and will be explained in detail in Section 3.1.2.

In the following, the data term and the smoothness term will be presented, before the details of the minimization, the necessary conditions and the discretization will be given in the next sections.

3.1.1 Data Term

The data term combines several ideas from before. First, it uses two constancy assumptions: the brightness constancy assumption and additionally the gradient constancy assumption as presented in Section 2.5.1. Both of them are used in their non-linearized form as described in Section 2.5.2. Additionally, they are both separately robustified using a penalizer function Ψ_D as explained in Section 2.6.1. This offers the advantage that the two assumptions are treated separately, since outliers can occur independently for the brightness and constancy assumption. When combining all these ideas, the resulting data term reads

$$D(\mathbf{u}) = \int_\Omega \Psi_D\left((f(\mathbf{x}+\mathbf{u}, t+1) - f(\mathbf{x}, t))^2\right) + \gamma \cdot \Psi_D\left(\|\nabla f(\mathbf{x}+\mathbf{u}, t+1) - \nabla f(\mathbf{x}, t)\|_2^2\right)\, d\mathbf{x}\ , \tag{3.2}$$

where the sub-quadratic Charbonnier function was used as a penalizer function, as described in Section 2.6.3.

3.1.2 Smoothness Term

The focus of the method by Maurer *et al.* [MSB17b] lies on the smoothness term. In order to afterwards derive their order-adaptive smoothness term, first an anisotropic coupling-based regularization is presented.

Anisotropic Coupling-based Regularization Before getting to the order-adaptive approach, an anisotropic coupling model will be presented. It was first introduced by Hafner *et al.* [HSW15] in the context of focus fusion and has been adapted by Maurer *et al.* [MSB17b] for optical flow estimation. The regularization term combines the second-order coupling model as presented in Section 2.7.2 with the idea of anisotropic regularization as shown in Section 2.7.1 and reads

$$\begin{aligned} R(\mathbf{u}) = \int_\Omega \inf_{\mathbf{a},\mathbf{b}} \Big\{ & \sum_{l=1}^{2} \Psi_l((\mathbf{r}_l^\top(\nabla u - \mathbf{a}))^2 + (\mathbf{r}_l^\top(\nabla v - \mathbf{b}))^2) \\ & + \beta \cdot \sum_{l=1}^{2} \Psi_l \left(\sum_{k=1}^{2} \left(\mathbf{r}_k^\top \mathcal{J}\mathbf{a}\,\mathbf{r}_l\right)^2 + \left(\mathbf{r}_k^\top \mathcal{J}\mathbf{b}\,\mathbf{r}_l\right)^2 \right) \Big\}\, d\mathbf{x}\,. \end{aligned} \tag{3.3}$$

In the following, this model will be combined further.

Order-Adaptive Smoothness Term As shown in another work by Maurer *et al.* [MSV+17], in some cases a first-order smoothness term is preferable while in other cases a second-order penalizer yields better results. A first-order regularization is well-suited for sequences with fronto-parallel motion, where the optical flow is mostly piecewise constant. On the other hand, for sequences with ego-motion of the camera, as e.g. in automotive applications, second order regularization provides better results since the resulting flow field is piecewise affine. In order to overcome the need to decide on the regularization order in advance, the method by Maurer *et al.* [MSB17b] strives to adaptively change the order as needed.

As bases for their method, they choose the anisotropic first-order regularization term as presented in Equation (2.90) and the anisotropic second-order coupling term as presented in Equation (3.3). Both terms are combined in a convex combination using a new order selection parameter $c \in [0, 1]$. Its goal is to adaptively determine the best order. Their base model is given as the functional

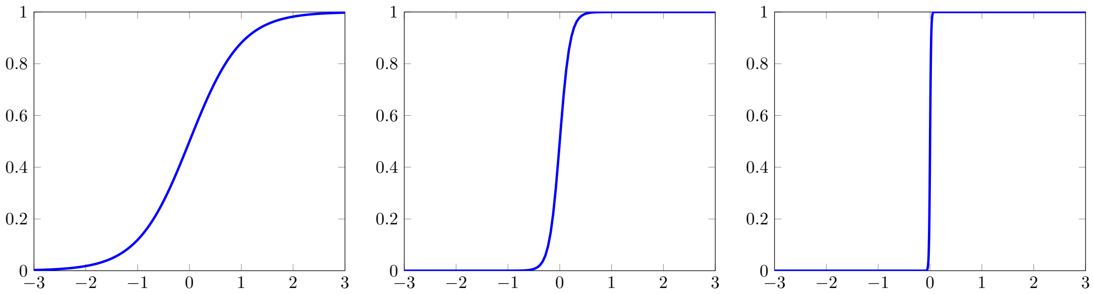
$$R(\mathbf{u}, c) = \int_\Omega \inf_{\mathbf{a},\mathbf{b}} \Big\{ c \cdot S_1(\mathbf{u}) + (1-c) \cdot S_2(\mathbf{u},\mathbf{a},\mathbf{b}) + \beta \cdot S_{\text{aux}}(\mathbf{a},\mathbf{b}) + \phi_\lambda(c) \Big\}\, d\mathbf{x}\,, \tag{3.4}$$

with

$$S_1(\mathbf{u}) = \sum_{l=1}^{2} \Psi_l((\mathbf{r}_l^\top \nabla u)^2 + (\mathbf{r}_l^\top \nabla v)^2)\,, \tag{3.5}$$

$$S_2(\mathbf{u},\mathbf{a},\mathbf{b}) = \sum_{l=1}^{2} \Psi_l \left((\mathbf{r}_l^\top(\nabla u - \mathbf{a}))^2 + (\mathbf{r}_l^\top(\nabla v - \mathbf{b}))^2 \right)\,, \tag{3.6}$$

$$S_{\text{aux}}(\mathbf{a},\mathbf{b}) = \sum_{l=1}^{2} \Psi_l \left(\sum_{k=1}^{2} \left(\mathbf{r}_k^\top \mathcal{J}\mathbf{a}\,\mathbf{r}_l\right)^2 + \left(\mathbf{r}_k^\top \mathcal{J}\mathbf{b}\,\mathbf{r}_l\right)^2 \right)\,. \tag{3.7}$$



(a) $\lambda = 0.5$ **(b)** $\lambda = 0.1$ **(c)** $\lambda = 0.01$

Figure 3.1: Sigmoid function for different values of λ with the threshold $T = 0$.

As introduced before, S_1 is an anisotropic first-order smoothness term and S_2 in conjunction with S_{aux} an indirect anisotropic second-order smoothness term. Note that S_1 and S_2 have a very similar structure, with the former enforcing the flow to be close to a zero flow field and the latter enforcing it to be close to the auxiliary flow field. The parameter c balances the two smoothness variants and is controlled by a selection term $\phi_\lambda(c)$. In the following, this selection term will be derived.

Derivation of Selection Term Whether the first or second-order penalization should be selected will be determined by looking at the energy difference of the two smoothness variants which are integrated over the image domain. Additionally, a small threshold T is added. This results in a minimum average benefit [MSB17b] that term S_2 needs to have over S_1. This is because the second-order coupling model implicitly includes the first-order model, since for $\mathbf{a} = \mathbf{b} = \mathbf{0}$, the coupling model is reduced exactly to the first-order model. Thus, the functions fitting the first-order smoothness assumption can be seen as a subset of the functions fitting the second-order smoothness assumption. The reason for an additional integration of the first-order regularization is to have an explicit combination of both orders, which provides better results, as shown by Maurer *et al.* [MSB17b]. The threshold thus prevents always selecting the second-order term.

The resulting integrated energy difference value Δ is fed into a sigmoid function, which is close to 1 for large values and close to zero for small values, yielding the definition

$$c = \frac{1}{1 + e^{-\Delta/\lambda}} \quad \text{with} \quad \Delta = T + \frac{1}{|\Omega|} \cdot \int_\Omega S_2 - S_1 \, d\mathbf{x} \,. \tag{3.8}$$

Figure 3.1 shows the sigmoid function plot for three different values of λ and for a value of $T = 0$. It becomes obvious that λ relates to the slope of the sigmoid function. Equation (3.8) yields a value close to 1 for c if term S_2 is larger than term S_1 over the image domain and supersedes the threshold T. Otherwise it approaches zero. Therefore, it can be used as a decision parameter between first and second order.

In the following, a selection term ϕ_λ will be derived with the goal of exactly modeling the desired behavior in Equation (3.8). First, Equation (3.8) can be further simplified as follows:

$$\frac{1}{c} - 1 = e^{-\Delta/\lambda} \tag{3.9}$$

$$\Leftrightarrow \lambda \cdot \ln\left(\frac{1}{c} - 1\right) = -\left(T + \frac{1}{|\Omega|} \cdot \int_\Omega S_2 - S_1 \, d\mathbf{x}\right) \tag{3.10}$$

$$\Leftrightarrow \int_\Omega S_1 - S_2 \, d\mathbf{x} = |\Omega| \left(\lambda \cdot \ln \left(\frac{1}{c} - 1 \right) + T \right) \ . \tag{3.11}$$

Since the optimal c should be a minimizer of R (see Equation (3.4)), its derivative should be zero:

$$\begin{aligned}
0 &= \frac{\partial R}{\partial c} && (3.12)\\
0 &= \int_\Omega S_1(\mathbf{u}) - S_2(\mathbf{u}, \mathbf{a}, \mathbf{b}) \, d\mathbf{x} + \int_\Omega \phi'_\lambda(c) \, d\mathbf{x} && (3.13)\\
0 &= |\Omega| \left(\lambda \cdot \ln \left(\frac{1}{c} - 1 \right) + T \right) + |\Omega| \cdot \phi'_\lambda(c) && (3.14)\\
\phi'_\lambda(c) &= -\lambda \cdot \ln \left(\frac{1}{c} - 1 \right) - T && (3.15)\\
\phi_\lambda(c) &= \lambda \cdot \left(\ln (1-c) - c \cdot \ln \left(\frac{1}{c} - 1 \right) \right) - Tc + C \ . && (3.16)
\end{aligned}$$

When choosing the integration constant $C = T$ and defining a function $\phi(c)$ as

$$\begin{aligned}
\phi(c) &:= \frac{\phi_\lambda(c) - (1-c) \cdot T}{\lambda} && (3.17)\\
&= \ln (1-c) - c \cdot \ln \left(\frac{1}{c} - 1 \right) && (3.18)\\
&= c \cdot \ln(c) + (1-c) \cdot \ln(1-c) \ , && (3.19)
\end{aligned}$$

such that

$$\phi_\lambda(c) = \lambda \cdot \phi(c) + (1-c) \cdot T \ , \tag{3.20}$$

it can be inserted into the regularization term and

$$R(\mathbf{u}, c) = \int_\Omega \inf_{\mathbf{a},\mathbf{b}} \left\{ \ c \ \cdot S_1(\mathbf{u}) + (1-c) \cdot S_2(\mathbf{u}, \mathbf{a}, \mathbf{b}) + \beta \cdot S_{\text{aux}}(\mathbf{a}, \mathbf{b}) + \phi_\lambda(c) \right\} d\mathbf{x} \tag{3.21}$$

can be rewritten as

$$R(\mathbf{u}, c) = \int_\Omega \inf_{\mathbf{a},\mathbf{b}} \left\{ c \cdot S_1(\mathbf{u}) + (1-c) \cdot (S_2(\mathbf{u}, \mathbf{a}, \mathbf{b}) + T) + \beta \cdot S_{\text{aux}}(\mathbf{a}, \mathbf{b}) + \lambda \cdot \phi(c) \right\} d\mathbf{x} \ , \tag{3.22}$$

which makes the dependency on λ and T explicit. The plot of $\phi(c)$ is shown in Figure 3.2.

Non-Local Isotropic Order Selection When defining c as in Equation (3.8), we are only able to estimate a single selection parameter per image. However, as Maurer *et al.* [MSB17b] describe, the best result can be achieved by choosing c as a spatially variant function $c(\mathbf{x})$ that defines c for every pixel and embedding it into a non-local scheme. This is done by determining a local order selection per pixel, which is then integrated over a local neighborhood $\mathcal{N}(\mathbf{x})$.

The resulting regularization term then reads:

$$R(\mathbf{u}, c) = \int_\Omega \inf_{\mathbf{a},\mathbf{b}} \left\{ \bar{c} \cdot S_1(\mathbf{u}) + (1-\bar{c}) \cdot (S_2(\mathbf{u}, \mathbf{a}, \mathbf{b}) + T) + \beta \cdot S_{\text{aux}}(\mathbf{a}, \mathbf{b}) + \lambda \cdot \phi(c) \right\} d\mathbf{x} \tag{3.23}$$

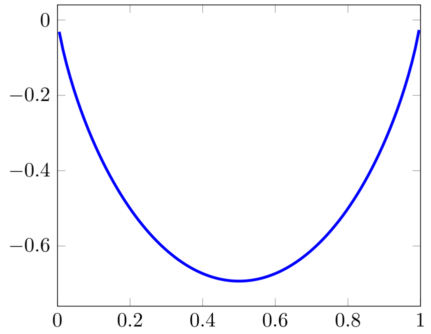


Figure 3.2: Plot of $\phi(c)$.

with

$$\bar{c}(\mathbf{x}) = \frac{1}{|\mathcal{N}(\mathbf{x})|} \int_{\mathcal{N}(\mathbf{x})} c(\mathbf{y}) \, d\mathbf{y} \; , \tag{3.24}$$

where

$$c(\mathbf{x}) = \frac{1}{1 + e^{-\Delta/\lambda}} \quad \text{with} \quad \Delta = \int_{\mathcal{N}(\mathbf{x})} \frac{1}{|\mathcal{N}(\mathbf{y})|} \cdot (T + S_2 - S_1) \, d\mathbf{y} \tag{3.25}$$

is the energy benefit within a small local neighborhood $\mathcal{N}(\mathbf{x})$. Since Maurer *et al.* [MSB17b] achieved the best results using this non-local approach, it will be solely used in the following.

3.2 Minimization

In order to minimize the energy functional as shown in Equation (3.1) using the data term from Equation (3.2) and the regularization term from Equation (3.23), many concepts need to be combined. The resulting optimization scheme then consists of the following nested iterations:

- *Warping Strategy*: The warping strategy computes $f(\mathbf{x}, t)$ and $f(\mathbf{x}, t+1)$ on the current resolution level and compensates $f(\mathbf{x}, t+1)$ by the already computed motion (*backward registration*). This changes the problem from non-linear and non-convex to a series of non-linear but (mostly) convex problems.
- *Alternating Minimization*: In order to jointly estimate the order selection map c and the flow $\mathbf{u}$, an alternating minimization strategy is introduced, which shifts between computing c, keeping the rest fixed and estimating the flow, keeping the order map fixed.
- *Lagged Non-Linearity*: The lagged non-linearity method is used since several sub-quadratic penalizer functions Ψ are used throughout the data and smoothness term. It introduces a new fixed-point iteration to compute the non-linear parts of the equations separately from its linear parts. This way a non-linear problem is reduced to a series of linear problems.
- *Inner Iteration*: Finally, the resulting linear system of equations is solved with the Jacobi, Gauss-Seidel or SOR method.

To derive these four nested iterations, the first step is giving a differential formulation for the energy. This is needed for the warping strategy. Then, the alternating minimization is explained. Afterwards the Euler-Lagrange equations are derived, which are finally discretized and solved.

3.2.1 Differential Formulation

Let k be the current warping iteration resolution level. The optical flow is formulated in a differential manner:

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{du}^k = (u^k, v^k)^\top + (du^k, dv^k)^\top \; . \tag{3.26}$$

Since $\mathbf{u}^k$ is taken from the previous resolution level, the goal is now to estimate the differential flow $\mathbf{du}^k$. Then, on each level k, the following differential energy functional must be minimized:

$$E(\mathbf{du}^k, c^k) = D(\mathbf{du}^k) + \alpha \cdot R(\mathbf{du}^k, c^k) \; . \tag{3.27}$$

In the following, first the differential data term and afterwards the regularization term will be derived.

Data Term Let f^k be the image data f resampled to the current resolution corresponding to k. Then the data term $D(\mathbf{du}^k)$ is given by:

$$\begin{aligned} D(\mathbf{du}^k) = \int_\Omega & \Psi_D \left(\left(f^k(\mathbf{x} + \mathbf{u}^k + \mathbf{du}^k, t+1) - f^k(\mathbf{x}, t) \right)^2 \right) \\ & + \gamma \cdot \Psi_D \left(\left\| \nabla f^k(\mathbf{x} + \mathbf{u}^k + \mathbf{du}^k, t+1) - \nabla f^k(\mathbf{x}, t) \right\|_2^2 \right) \, d\mathbf{x} \; . \end{aligned} \tag{3.28}$$

In order to extract $\mathbf{du}^k$ from the argument of the function, the following expressions are linearized w.r.t. $\mathbf{du}^k$:

$$f^k(\mathbf{x}{+}\mathbf{u}^k{+}\mathbf{du}^k, t{+}1) \approx f^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) + f_x^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) \cdot du^k + f_y^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) \cdot dv^k \; , \tag{3.29}$$

$$f_x^k(\mathbf{x}{+}\mathbf{u}^k{+}\mathbf{du}^k, t{+}1) \approx f_x^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) + f_{xx}^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) \cdot du^k + f_{xy}^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) \cdot dv^k \; , \tag{3.30}$$

$$f_y^k(\mathbf{x}{+}\mathbf{u}^k{+}\mathbf{du}^k, t{+}1) \approx f_y^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) + f_{yx}^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) \cdot du^k + f_{yy}^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) \cdot dv^k \; . \tag{3.31}$$

Here, $f^k(\mathbf{x} + \mathbf{u}^k, t+1)$ is computed using backward registration, i.e. warping the second image by the already computed flow $\mathbf{u}^k$ on resolution level k. After plugging in the linearized expressions, the differential data term reads:

$$\begin{aligned} & D(\mathbf{du}^k) = \\ & \int_\Omega \Psi_D \left((f^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) {+} f_x^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) \cdot du^k {+} f_y^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) \cdot dv^k {-} f^k(\mathbf{x}, t))^2 \right) \\ & \quad {+}\gamma \cdot \Psi_D \Big(\big(f_x^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) {+} f_{xx}^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) \cdot du^k {+} f_{xy}^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) \cdot dv^k {-} f_x^k(\mathbf{x}, t) \big)^2 \\ & \quad\quad + \big(f_y^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) {+} f_{yx}^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) \cdot du^k {+} f_{yy}^k(\mathbf{x}{+}\mathbf{u}^k, t{+}1) \cdot dv^k {-} f_y^k(\mathbf{x}, t) \big)^2 \Big) \, d\mathbf{x} \; . \end{aligned} \tag{3.32}$$

As in Section 2.5.2, the motion tensor abbreviation allows for a more compact notation. For the brightness and gradient constancy assumption, two motion tensors, $\mathbf{J}^k_{\mathrm{bca}}$ and $\mathbf{J}^k_{\mathrm{gca}}$, are defined:

$$\mathbf{J}^k_{\mathrm{bca}} = \begin{pmatrix} f^k_x(\mathbf{x}+\mathbf{u}^k, t+1) \\ f^k_y(\mathbf{x}+\mathbf{u}^k, t+1) \\ f^k(\mathbf{x}+\mathbf{u}^k, t+1) - f^k(\mathbf{x}, t) \end{pmatrix} \cdot \begin{pmatrix} f^k_x(\mathbf{x}+\mathbf{u}^k, t+1) \\ f^k_y(\mathbf{x}+\mathbf{u}^k, t+1) \\ f^k(\mathbf{x}+\mathbf{u}^k, t+1) - f^k(\mathbf{x}, t) \end{pmatrix}^\top , \tag{3.33}$$

$$\mathbf{J}^k_{\mathrm{gca}} = \mathbf{J}^k_{\mathrm{gca},x} + \mathbf{J}^k_{\mathrm{gca},y} \tag{3.34}$$

with

$$\mathbf{J}^k_{\mathrm{gca},x} = \begin{pmatrix} f^k_{xx}(\mathbf{x}+\mathbf{u}^k, t+1) \\ f^k_{xy}(\mathbf{x}+\mathbf{u}^k, t+1) \\ f^k_x(\mathbf{x}+\mathbf{u}^k, t+1) - f^k_x(\mathbf{x}, t) \end{pmatrix} \cdot \begin{pmatrix} f^k_{xx}(\mathbf{x}+\mathbf{u}^k, t+1) \\ f^k_{xy}(\mathbf{x}+\mathbf{u}^k, t+1) \\ f^k_x(\mathbf{x}+\mathbf{u}^k, t+1) - f^k_x(\mathbf{x}, t) \end{pmatrix}^\top , \tag{3.35}$$

$$\mathbf{J}^k_{\mathrm{gca},y} = \begin{pmatrix} f^k_{yx}(\mathbf{x}+\mathbf{u}^k, t+1) \\ f^k_{yy}(\mathbf{x}+\mathbf{u}^k, t+1) \\ f^k_y(\mathbf{x}+\mathbf{u}^k, t+1) - f^k_y(\mathbf{x}, t) \end{pmatrix} \cdot \begin{pmatrix} f^k_{yx}(\mathbf{x}+\mathbf{u}^k, t+1) \\ f^k_{yy}(\mathbf{x}+\mathbf{u}^k, t+1) \\ f^k_y(\mathbf{x}+\mathbf{u}^k, t+1) - f^k_y(\mathbf{x}, t) \end{pmatrix}^\top . \tag{3.36}$$

Additionally, similar to before, the differential flow is parametrized as

$$\mathbf{dw}^k = \left(\left(\mathbf{du}^k\right)^\top, 1 \right)^\top = \left(du^k, dv^k, 1 \right)^\top . \tag{3.37}$$

Then the data term in motion tensor notation reads

$$D(\mathbf{du}^k) = \int_\Omega \Psi_D \left((\mathbf{dw}^k)^\top \mathbf{J}^k_{\mathrm{bca}} \, \mathbf{dw}^k \right) + \gamma \cdot \Psi_D \left((\mathbf{dw}^k)^\top \mathbf{J}^k_{\mathrm{gca}} \, \mathbf{dw}^k \right) \, d\mathbf{x} \, . \tag{3.38}$$

Regularization Term The differential formulation of the regularization term is straightforward. It reads

$$\begin{aligned} R(\mathbf{du}^k, c^k) = \int_\Omega \inf_{\mathbf{da}^k, \mathbf{db}^k} \Big\{ & \bar{c}^k \cdot S_1(\mathbf{u}^k + \mathbf{du}^k) \\ & + (1 - \bar{c}^k) \cdot \Big(S_2(\mathbf{u}^k + \mathbf{du}^k, \mathbf{a}^k + \mathbf{da}^k, \mathbf{b}^k + \mathbf{db}^k) + T \Big) \\ & + \beta \cdot S_{\mathrm{aux}}(\mathbf{a}^k + \mathbf{da}^k, \mathbf{b}^k + \mathbf{db}^k) + \lambda \cdot \phi(c^k) \Big\} \, d\mathbf{x} \, , \end{aligned} \tag{3.39}$$

with

$$S_1(\mathbf{u}^k + \mathbf{du}^k) = \sum_{l=1}^{2} \Psi_l \bigg(\Big(\mathbf{r}_l^\top \nabla (u^k + du^k) \Big)^2 + \Big(\mathbf{r}_l^\top \nabla (v^k + dv^k) \Big)^2 \bigg) , \tag{3.40}$$

$$\begin{aligned} S_2(\mathbf{u}^k + \mathbf{du}^k, \mathbf{a}^k + \mathbf{da}^k, \mathbf{b}^k + \mathbf{db}^k) = \sum_{l=1}^{2} \Psi_l \bigg(& \Big(\mathbf{r}_l^\top \Big(\nabla (u^k + du^k) - (\mathbf{a}^k + \mathbf{da}^k) \Big) \Big)^2 \\ & + \Big(\mathbf{r}_l^\top \Big(\nabla (v^k + dv^k) - (\mathbf{b}^k + \mathbf{db}^k) \Big) \Big)^2 \bigg) , \end{aligned} \tag{3.41}$$

$$S_{\mathrm{aux}}(\mathbf{a}^k{+}\mathbf{da}^k,\mathbf{b}^k{+}\mathbf{db}^k) = \sum_{l=1}^{2}\Psi_l\Bigg(\sum_{k=1}^{2}\Big(\mathbf{r}_k^\top\mathcal{J}(\mathbf{a}^k{+}\mathbf{da}^k)\,\mathbf{r}_l\Big)^2 + \Big(\mathbf{r}_k^\top\mathcal{J}(\mathbf{b}^k{+}\mathbf{db}^k)\,\mathbf{r}_l\Big)^2\Bigg)\ . \tag{3.42}$$

This concludes all steps needed for the first iterative scheme introduced by the warping strategy.

3.2.2 Alternating Minimization

The second iteration as introduced at the beginning of Section 3.2 is the alternating minimization step, which shifts between computing c^{k+1} given $\mathbf{du}^k, \mathbf{da}^k, \mathbf{db}^k$ and computing $\mathbf{du}^{k+1}, \mathbf{da}^{k+1}, \mathbf{db}^{k+1}$ given c^{k+1}:

$$c^{k+1} = \arg\min_{c} E(\mathbf{du}^k, \mathbf{da}^k, \mathbf{db}^k, c)\ , \tag{3.43}$$

$$\mathbf{du}^{k+1}, \mathbf{da}^{k+1}, \mathbf{db}^{k+1} = \arg\min_{\mathbf{du},\mathbf{da},\mathbf{db}} E(\mathbf{du}, \mathbf{da}, \mathbf{db}, c^{k+1})\ . \tag{3.44}$$

In the following, first the minimization w.r.t. the order selection parameter will be derived. Afterwards the minimization w.r.t. the flow will be determined.

3.2.2.1 Minimization w.r.t. the Order Selection Map

When the energy is minimized w.r.t. c, it can be replaced with a version that drops all terms not containing c. The resulting energy functional E_c is given by:

$$E_c(c) = \alpha \cdot \int_\Omega \inf_{\mathbf{da}^k,\mathbf{db}^k} \Big\{\ \bar{c}\cdot S_1(\mathbf{u}^k + \mathbf{du}^k) + (1-\bar{c})\cdot\Big(S_2(\mathbf{u}^k+\mathbf{du}^k, \mathbf{a}^k+\mathbf{da}^k, \mathbf{b}^k+\mathbf{db}^k) + T\Big) + \lambda\cdot\phi(c)\Big\}\ d\mathbf{x}\ . \tag{3.45}$$

Before calculating the associated Euler-Lagrange equations, it is important to note that in the current formulation, c is not defined pointwise inside the integral due to the calculation of $\bar{c}$. Thus, in order to make use of the standard case of deriving Euler-Lagrange equations given by Equations (2.4) to (2.7), the term needs to be reformulated.

Reformulation In order to explain the reformulation, in the following a simplified 1D variant of the energy functional shown in Equation (3.45) will be used:

$$R(c) = \int_\Omega \bar{c}\cdot S(x)\ dx \tag{3.46}$$

$$= \int_\Omega \left(\frac{1}{|\mathcal{N}(x)|}\int_{\mathcal{N}(x)} c(y)\,dy\right)\cdot S(x)\ dx\ . \tag{3.47}$$

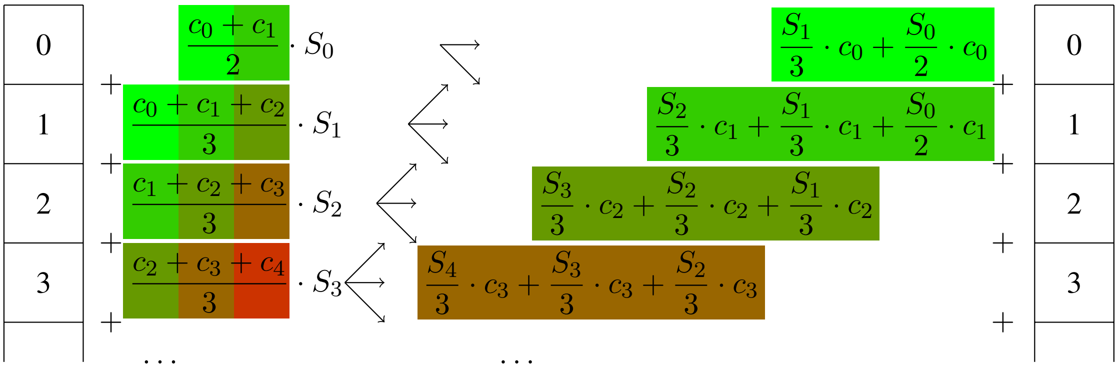

0
1
2
3
$\frac{c_0+c_1}{2}\cdot S_0$
$\frac{c_0+c_1+c_2}{3}\cdot S_1$
$\frac{c_1+c_2+c_3}{3}\cdot S_2$
$\frac{c_2+c_3+c_4}{3}\cdot S_3$
+
...
$\frac{S_1}{3}\cdot c_0+\frac{S_0}{2}\cdot c_0$
$\frac{S_2}{3}\cdot c_1+\frac{S_1}{3}\cdot c_1+\frac{S_0}{2}\cdot c_1$
$\frac{S_3}{3}\cdot c_2+\frac{S_2}{3}\cdot c_2+\frac{S_1}{3}\cdot c_2$
$\frac{S_4}{3}\cdot c_3+\frac{S_3}{3}\cdot c_3+\frac{S_2}{3}\cdot c_3$
...
0
1
2
3

Figure 3.3: Schematic illustration of the reformulation process for a pointwise definition. The left-hand side shows the original formulation with the c_i's being distributed over different parts of the sum. After rearrangement on the right-hand side, all sum parts contain the same c_i.

Its discrete counterpart reads

$$R(c) = \sum_{x\in\Omega}\left(\frac{1}{|\mathcal{N}(x)|}\cdot\sum_{y\in\mathcal{N}(x)} c_y\right)\cdot S_x\ . \tag{3.48}$$

After writing out the complete sums and rearranging the term, it becomes obvious that the expression can be reformulated as follows:

$$R(c) = \sum_{x\in\Omega}\left(c_x\cdot\sum_{y\in\mathcal{N}(x)}\frac{1}{|\mathcal{N}(y)|}\cdot S_y\right)\ . \tag{3.49}$$

Figure 3.3 illustrates this rearrangement for a neighborhood of size 3: The complete sum is sorted by the c_i's. After the reformulation, all c's appear pointwise in the outer sum.

In the same way, the continuous formulation in Equation (3.47) can be reformulated. It then reads

$$R(c) = \int_\Omega c(\mathbf{x})\cdot\left(\int_{\mathcal{N}(\mathbf{x})}\frac{1}{|\mathcal{N}(\mathbf{y})|}\cdot S(\mathbf{y})\ d\mathbf{y}\right)\ d\mathbf{x}\ . \tag{3.50}$$

Finally, the energy functional to compute c (see Equation (3.45)) can be reformulated in the same way:

$$\begin{aligned} E_c(c) = \alpha\cdot\int_\Omega \inf_{\mathbf{da}^k,\mathbf{db}^k}\Big\{\ & \bar{c}\cdot S_1 \\ & +(1-\bar{c})\cdot(S_2+T) \\ & +\lambda\cdot\phi(c)\Big\}\ d\mathbf{x} \end{aligned} \tag{3.51}$$

$$\begin{aligned}
&= \alpha \cdot \int_\Omega \inf_{\mathbf{da}^k, \mathbf{db}^k} \Bigg\{ \left(\frac{1}{|\mathcal{N}(\mathbf{x})|} \int_{\mathcal{N}(\mathbf{x})} c(\mathbf{y}) \, d\mathbf{y} \right) \cdot S_1 \\
&\qquad + S_2 + T - \left(\frac{1}{|\mathcal{N}(\mathbf{x})|} \int_{\mathcal{N}(\mathbf{x})} c(\mathbf{y}) \, d\mathbf{y} \right) \cdot (S_2 + T) \\
&\qquad + \lambda \cdot \phi(c) \Bigg\} \, d\mathbf{x} \qquad (3.52) \\
&= \alpha \cdot \int_\Omega \inf_{\mathbf{da}^k, \mathbf{db}^k} \Bigg\{ c(\mathbf{x}) \cdot \left(\int_{\mathcal{N}(\mathbf{x})} \frac{1}{|\mathcal{N}(\mathbf{y})|} \, S_1 \, d\mathbf{y} \right) \\
&\qquad + S_2 + T - c(\mathbf{x}) \cdot \left(\int_{\mathcal{N}(\mathbf{x})} \frac{1}{|\mathcal{N}(\mathbf{y})|} \, (S_2 + T) \; d\mathbf{y} \right) \\
&\qquad + \lambda \cdot \phi(c) \Bigg\} \, d\mathbf{x} \; . \qquad (3.53)
\end{aligned}$$

Note that now all occurrences of c inside the integral have the integration variable $\mathbf{x}$ as an argument: $c(\mathbf{x})$. Hence we can resort to the standard case of deriving Euler-Lagrange equations as given by Equations (2.4) to (2.7).

Euler-Lagrange Equation Since c is now defined pointwise inside the integral, the Euler-Lagrange equation can be stated. The energy functional now has the proper form:

$$E_c(c) = \int_\Omega F(\mathbf{x}, c) \, d\mathbf{x} \; . \qquad (3.54)$$

Since no derivatives of c are present, the Euler-Lagrange equation simply reads

$$F_c = 0 \; , \qquad (3.55)$$

with no additional boundary condition. The necessary condition is hence given by

$$\begin{aligned}
0 &= F_c \qquad (3.56) \\
&= \int_{\mathcal{N}(\mathbf{x})} \frac{1}{|\mathcal{N}(\mathbf{y})|} \, S_1 \, d\mathbf{y} - \int_{\mathcal{N}(\mathbf{x})} \frac{1}{|\mathcal{N}(\mathbf{y})|} \, (S_2 + T) \; d\mathbf{y} \quad + \lambda \cdot \phi'(c(\mathbf{x})) \; . \qquad (3.57)
\end{aligned}$$

Rearranging terms yields

$$\lambda \cdot \phi'(c(\mathbf{x})) = \int_{\mathcal{N}(\mathbf{x})} \frac{1}{|\mathcal{N}(\mathbf{y})|} \cdot (T + S_2 - S_1) \; d\mathbf{y} =: \Delta \; . \qquad (3.58)$$

The derivative of $\phi(c)$ is given as $\phi'(c) = -\ln(\frac{1}{c} - 1)$. With this, the necessary condition is further simplified:

$$\begin{aligned}
&\Leftrightarrow -\lambda \cdot \ln \left(\frac{1}{c} - 1 \right) = \Delta \qquad (3.59) \\
&\Leftrightarrow \frac{1}{c} - 1 = e^{-\Delta/\lambda} \qquad (3.60) \\
&\Leftrightarrow c = \frac{1}{1 + e^{-\Delta/\lambda}} \; . \qquad (3.61)
\end{aligned}$$

This is now exactly the definition of c from Equation (3.25). This is not surprising, since the selection term $\phi(c)$ was modeled via Equation (3.19) in such a way that we obtained this definition. Moreover, this also shows that in contrast to the minimization of the flow, the minimizer of c can be computed explicitly. Thus, the alternating step of computing c comes down to a single calculation which can be done together with computing the non-linear terms for the lagged non-linearity method.

3.2.2.2 Minimization w.r.t. the Flow

So far, from the alternating steps of computing c and the flow, only the minimization w.r.t. the order map was derived. In the following, the Euler-Lagrange equations of minimizing the differential energy functional w.r.t. the flow variables will be presented.

The energy functional to compute the minimizing flow field has the form

$$\begin{aligned} E(\mathbf{du}, \mathbf{da}, \mathbf{db}) &= \int_\Omega F(\mathbf{x}, \mathbf{du}, \mathbf{da}, \mathbf{db}, \mathcal{J}\mathbf{du}, \mathcal{J}\mathbf{da}, \mathcal{J}\mathbf{db})\; d\mathbf{x} && (3.62) \\ &= \int_\Omega F(x, y, du, dv, da_1, da_2, db_1, db_2, du_x, du_y, dv_x, dv_y, \\ &\qquad da_{1x}, da_{1y}, da_{2x}, da_{2y}, db_{1x}, db_{1y}, db_{2x}, db_{2y})\; dx\, dy && (3.63) \end{aligned}$$

using the following definitions:

$$\mathbf{x} = (x, y)^\top,\ \mathbf{du} = (du, dv)^\top,\ \mathbf{da} = (da_1, da_2)^\top,\ \mathbf{db} = (db_1, db_2)^\top\ . \tag{3.64}$$

In total, there are six minimality conditions for the flow and the coupling variables:

$$F_{du} - \frac{\partial}{\partial x} F_{du_x} - \frac{\partial}{\partial y} F_{du_y} = 0\ , \tag{3.65}$$

$$F_{dv} - \frac{\partial}{\partial x} F_{dv_x} - \frac{\partial}{\partial y} F_{dv_y} = 0\ , \tag{3.66}$$

$$F_{da_1} - \frac{\partial}{\partial x} F_{da_{1x}} - \frac{\partial}{\partial y} F_{da_{1y}} = 0\ , \tag{3.67}$$

$$F_{da_2} - \frac{\partial}{\partial x} F_{da_{2x}} - \frac{\partial}{\partial y} F_{da_{2y}} = 0\ , \tag{3.68}$$

$$F_{db_1} - \frac{\partial}{\partial x} F_{db_{1x}} - \frac{\partial}{\partial y} F_{db_{1y}} = 0\ , \tag{3.69}$$

$$F_{db_2} - \frac{\partial}{\partial x} F_{db_{2x}} - \frac{\partial}{\partial y} F_{db_{2y}} = 0\ . \tag{3.70}$$

The boundary conditions read:

$$\mathbf{n}^\top \begin{pmatrix} F_{du_x} \\ F_{du_y} \end{pmatrix} = 0\ , \qquad \mathbf{n}^\top \begin{pmatrix} F_{dv_x} \\ F_{dv_y} \end{pmatrix} = 0\ , \tag{3.71}$$

$$\mathbf{n}^\top \begin{pmatrix} F_{da_{1x}} \\ F_{da_{1y}} \end{pmatrix} = 0\ , \qquad \mathbf{n}^\top \begin{pmatrix} F_{da_{2x}} \\ F_{da_{2y}} \end{pmatrix} = 0\ , \tag{3.72}$$

$$\mathbf{n}^\top \begin{pmatrix} F_{db_{1x}} \\ F_{db_{1y}} \end{pmatrix} = 0\ , \qquad \mathbf{n}^\top \begin{pmatrix} F_{db_{2x}} \\ F_{db_{2y}} \end{pmatrix} = 0\ . \tag{3.73}$$

Definitions In order to state the necessary conditions, the following abbreviations are helpful for a more compact notation.

When using a sub-quadratic function Ψ, the inner derivative results in a multiplier in the Euler-Lagrange equation. When using the lagged non-linearity method, this multiplier is treated as constant. Thus, it makes sense to introduce the following abbreviations for several occurrences of inner derivatives:

$$\Psi'_{D,\mathrm{bca}} = \Psi'_D\left((\mathbf{dw}^k)^\top\, \mathbf{J}^k_{\mathrm{bca}}\, \mathbf{dw}^k\right) \ , \tag{3.74}$$

$$\Psi'_{D,\mathrm{gca}} = \Psi'_D\left((\mathbf{dw}^k)^\top\, \mathbf{J}^k_{\mathrm{gca}}\, \mathbf{dw}^k\right) \ , \tag{3.75}$$

$$\Psi'_{l,S_1} = \Psi'_l\Big((\mathbf{r}_l^\top \nabla(u^k + du^k))^2 + (\mathbf{r}_l^\top \nabla(v^k + dv^k))^2\Big) \ , \tag{3.76}$$

$$\Psi'_{l,S_2} = \Psi'_l\Big(\left(\mathbf{r}_l^\top\left(\nabla(u^k{+}du^k){-}(\mathbf{a}^k{+}\mathbf{da}^k)\right)\right)^2{+}\left(\mathbf{r}_l^\top\left(\nabla(v^k{+}dv^k){-}(\mathbf{b}^k{+}\mathbf{db}^k)\right)\right)^2\Big) \ , \tag{3.77}$$

$$\Psi'_{l,S_\mathrm{aux}} = \Psi'_l\left(\sum_{k=1}^{2}\left(\mathbf{r}_k^\top \mathcal{J}(\mathbf{a}^k + \mathbf{da}^k)\,\mathbf{r}_l\right)^2 + \left(\mathbf{r}_k^\top \mathcal{J}(\mathbf{b}^k + \mathbf{db}^k)\,\mathbf{r}_l\right)^2\right) \ . \tag{3.78}$$

Moreover, the Euler-Lagrange equations contain divergence expressions where it makes sense to introduce the following diffusion tensor definitions:

$$\mathbf{T}_{S_1} = \bar{c}\cdot\sum_{l=1}^{2}\Psi'_{l,S_1}\cdot\mathbf{r}_l\cdot\mathbf{r}_l^\top \ , \tag{3.79}$$

$$\mathbf{T}_{S_2} = (1-\bar{c})\cdot\sum_{l=1}^{2}\Psi'_{l,S_2}\cdot\mathbf{r}_l\cdot\mathbf{r}_l^\top \ , \tag{3.80}$$

$$\mathbf{T}_{S_\mathrm{aux}} = \sum_{l=1}^{2}\Psi'_{l,S_\mathrm{aux}}\cdot\mathbf{r}_l\cdot\mathbf{r}_l^\top \ . \tag{3.81}$$

Flow Minimality Conditions In order to determine the first two Euler-Lagrange equations, the necessary derivatives are calculated. The derivatives F_{du} and F_{dv} are given as:

$$F_{du} = \frac{\partial}{\partial du}D(\mathbf{du}^k) + \alpha\cdot\frac{\partial}{\partial du}R(\mathbf{du}^k) = \frac{\partial}{\partial du}D(\mathbf{du}^k) \tag{3.82}$$

$$\begin{aligned} &= \Psi'_{D,\mathrm{bca}}\cdot 2\cdot\left(\mathbf{J}^k_{\mathrm{bca},11}\cdot du^k + \mathbf{J}^k_{\mathrm{bca},12}\cdot dv^k + \mathbf{J}^k_{\mathrm{bca},13}\right)\\ &\quad + \gamma\cdot\Psi'_{D,\mathrm{gca}}\cdot 2\cdot\left(\mathbf{J}^k_{\mathrm{gca},11}\cdot du^k + \mathbf{J}^k_{\mathrm{gca},12}\cdot dv^k + \mathbf{J}^k_{\mathrm{gca},13}\right) \ , \end{aligned} \tag{3.83}$$

$$\begin{aligned} F_{dv} &= \Psi'_{D,\mathrm{bca}}\cdot 2\cdot\left(\mathbf{J}^k_{\mathrm{bca},21}\cdot du^k + \mathbf{J}^k_{\mathrm{bca},22}\cdot dv^k + \mathbf{J}^k_{\mathrm{bca},23}\right)\\ &\quad + \gamma\cdot\Psi'_{D,\mathrm{gca}}\cdot 2\cdot\left(\mathbf{J}^k_{\mathrm{gca},21}\cdot du^k + \mathbf{J}^k_{\mathrm{gca},22}\cdot dv^k + \mathbf{J}^k_{\mathrm{gca},23}\right) \ . \end{aligned} \tag{3.84}$$

The derivatives $F_{du_x}, F_{du_y}, F_{dv_x}$ and F_{dv_y} read:

$$\begin{aligned}
F_{du_x} &= \alpha \cdot \left(\bar{c} \cdot \frac{\partial}{\partial du_x} S_1(\mathbf{u}^k + \mathbf{du}^k) \right. \\
&\quad \left. + (1-\bar{c}) \cdot \frac{\partial}{\partial du_x} S_2(\mathbf{u}^k + \mathbf{du}^k, \mathbf{a}^k + \mathbf{da}^k, \mathbf{b}^k + \mathbf{db}^k) \right) && (3.85) \\
&= 2\alpha \sum_{l=1}^{2} \bar{c}\, \Psi'_{l,S_1} \cdot \Big(\mathbf{r}_l^\top \nabla(u^k{+}du^k) \Big)\, r_{l1} \\
&\quad + (1-\bar{c})\, \Psi'_{l,S_2} \cdot \Big(\mathbf{r}_l^\top \Big(\nabla(u^k{+}du^k) {-} (\mathbf{a}^k{+}\mathbf{da}^k) \Big) \Big)\, r_{l1} \;, && (3.86) \\
F_{du_y} &= 2\alpha \sum_{l=1}^{2} \bar{c}\, \Psi'_{l,S_1} \cdot \Big(\mathbf{r}_l^\top \nabla(u^k{+}du^k) \Big)\, r_{l2} \\
&\quad + (1-\bar{c})\, \Psi'_{l,S_2} \cdot \Big(\mathbf{r}_l^\top \Big(\nabla(u^k{+}du^k) {-} (\mathbf{a}^k{+}\mathbf{da}^k) \Big) \Big)\, r_{l2} \;, && (3.87) \\
F_{dv_x} &= 2\alpha \sum_{l=1}^{2} \bar{c}\, \Psi'_{l,S_1} \cdot \Big(\mathbf{r}_l^\top \nabla(v^k{+}dv^k) \Big)\, r_{l1} \\
&\quad + (1-\bar{c})\, \Psi'_{l,S_2} \cdot \Big(\mathbf{r}_l^\top \Big(\nabla(v^k{+}dv^k) {-} (\mathbf{b}^k{+}\mathbf{db}^k) \Big) \Big)\, r_{l1} \;, && (3.88) \\
F_{dv_y} &= 2\alpha \sum_{l=1}^{2} \bar{c}\, \Psi'_{l,S_1} \cdot \Big(\mathbf{r}_l^\top \nabla(v^k{+}dv^k) \Big)\, r_{l2} \\
&\quad + (1-\bar{c})\, \Psi'_{l,S_2} \cdot \Big(\mathbf{r}_l^\top \Big(\nabla(v^k{+}dv^k) {-} (\mathbf{b}^k{+}\mathbf{db}^k) \Big) \Big)\, r_{l2} \;. && (3.89)
\end{aligned}$$

Finally, following the definitions of Equations (3.65) and (3.66) and plugging the corresponding derivatives together yields the Euler-Lagrange equations for the flow estimation:

$$\begin{aligned}
0 &= \Psi'_{D,\mathrm{bca}} \cdot \Big(\mathbf{J}^k_{\mathrm{bca},11} \cdot du^k + \mathbf{J}^k_{\mathrm{bca},12} \cdot dv^k + \mathbf{J}^k_{\mathrm{bca},13} \Big) \\
&\quad + \gamma \cdot \Psi'_{D,\mathrm{gca}} \cdot \Big(\mathbf{J}^k_{\mathrm{gca},11} \cdot du^k + \mathbf{J}^k_{\mathrm{gca},12} \cdot dv^k + \mathbf{J}^k_{\mathrm{gca},13} \Big) \\
&\quad - \alpha \cdot \mathrm{div} \Big(\mathbf{T}_{S_1} \cdot \nabla(u^k + du^k) \Big) - \alpha \cdot \mathrm{div} \Big(\mathbf{T}_{S_2} \cdot \Big(\nabla(u^k + du^k) - (\mathbf{a}^k + \mathbf{da}^k) \Big) \Big) \;\;, && (3.90) \\
0 &= \Psi'_{D,\mathrm{bca}} \cdot \Big(\mathbf{J}^k_{\mathrm{bca},21} \cdot du^k + \mathbf{J}^k_{\mathrm{bca},22} \cdot dv^k + \mathbf{J}^k_{\mathrm{bca},23} \Big) \\
&\quad + \gamma \cdot \Psi'_{D,\mathrm{gca}} \cdot \Big(\mathbf{J}^k_{\mathrm{gca},21} \cdot du^k + \mathbf{J}^k_{\mathrm{gca},22} \cdot dv^k + \mathbf{J}^k_{\mathrm{gca},23} \Big) \\
&\quad - \alpha \cdot \mathrm{div} \Big(\mathbf{T}_{S_1} \cdot \nabla(v^k + dv^k) \Big) - \alpha \cdot \mathrm{div} \Big(\mathbf{T}_{S_2} \cdot \Big(\nabla(v^k + dv^k) - (\mathbf{b}^k + \mathbf{db}^k) \Big) \Big) \;\;. && (3.91)
\end{aligned}$$

The corresponding boundary conditions read:

$$\begin{aligned}
\mathbf{n}^\top \cdot \Big(\mathbf{T}_{S_1} \cdot \nabla(u^k + du^k) + \mathbf{T}_{S_2} \cdot \Big(\nabla(u^k + du^k) - (\mathbf{a}^k + \mathbf{da}^k) \Big) \Big) &= 0 \;, && (3.92) \\
\mathbf{n}^\top \cdot \Big(\mathbf{T}_{S_1} \cdot \nabla(v^k + dv^k) + \mathbf{T}_{S_2} \cdot \Big(\nabla(v^k + dv^k) - (\mathbf{b}^k + \mathbf{db}^k) \Big) \Big) &= 0 \;. && (3.93)
\end{aligned}$$

The resulting Euler-Lagrange equations are non-linear in $\mathbf{du}$, with the non-linear parts being $\Psi'_{D,\mathrm{bca}}$, $\Psi'_{D,\mathrm{gca}}$, $\mathbf{T}_{S_1}$ and $\mathbf{T}_{S_2}$.

Coupling Variables Minimality Conditions For the auxiliary variables the derivatives are:

$$F_{\mathbf{da}} = \alpha \cdot (1-\bar{c}) \cdot \frac{\partial}{\partial \mathbf{da}} S_2(\mathbf{u}^k + \mathbf{du}^k, \mathbf{a}^k + \mathbf{da}^k, \mathbf{b}^k + \mathbf{db}^k) \tag{3.94}$$

$$= -2\alpha \cdot (1-\bar{c}) \cdot \sum_{l=1}^{2} \Psi'_{l,S_2} \cdot \mathbf{r}_l \cdot \mathbf{r}_l^\top \left(\nabla(u^k + du^k) - (\mathbf{a}^k + \mathbf{da}^k)\right) \tag{3.95}$$

$$= -2\alpha \cdot \mathbf{T}_{S_2} \left(\nabla(u^k + du^k) - (\mathbf{a}^k + \mathbf{da}^k)\right) . \tag{3.96}$$

Before computing the other derivatives, the term S_{aux} can be reformulated in the following way:

$$S_{\text{aux}}(\mathbf{a}^k + \mathbf{da}^k, \mathbf{b}^k + \mathbf{db}^k) = \sum_{l=1}^{2} \Psi_l \left(\sum_{k=1}^{2} \left(\mathbf{r}_k^\top \mathcal{J}(\mathbf{a}^k + \mathbf{da}^k)\, \mathbf{r}_l\right)^2 + \left(\mathbf{r}_k^\top \mathcal{J}(\mathbf{b}^k + \mathbf{db}^k)\, \mathbf{r}_l\right)^2 \right) \tag{3.97}$$

$$= \sum_{l=1}^{2} \Psi_l \left(\left\| \mathcal{J}(\mathbf{a}^k + \mathbf{da}^k)\, \mathbf{r}_l \right\|^2 + \left\| \mathcal{J}(\mathbf{b}^k + \mathbf{db}^k)\, \mathbf{r}_l \right\|^2 \right) . \tag{3.98}$$

This transformation can be done since for two orthonormal vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ and an arbitrary vector $\mathbf{v}$, the following holds:

$$\sum_{k=1}^{2} \left(\mathbf{r}_k^\top \mathbf{v}\right)^2 = \mathbf{v}^\top \cdot \sum_{k=1}^{2} \mathbf{r}_k \mathbf{r}_k^\top \cdot \mathbf{v} = \mathbf{v}^\top \cdot \mathbf{I} \cdot \mathbf{v} = \|\mathbf{v}\|^2 . \tag{3.99}$$

The further derivatives are then given as

$$F_{da_{1x}} = \alpha\beta \cdot \frac{\partial}{\partial da_{1x}} S_{\text{aux}}(\mathbf{a}^k + \mathbf{da}^k, \mathbf{b}^k + \mathbf{db}^k) \tag{3.100}$$

$$= 2\alpha\beta \cdot \sum_{l=1}^{2} \Psi'_{l,S_{\text{aux}}} \cdot r_{l1} \cdot \mathbf{r}_l^\top \cdot \nabla(a_1^k + da_1^k) , \tag{3.101}$$

$$F_{da_{1y}} = 2\alpha\beta \cdot \sum_{l=1}^{2} \Psi'_{l,S_{\text{aux}}} \cdot r_{l2} \cdot \mathbf{r}_l^\top \cdot \nabla(a_1^k + da_1^k) , \tag{3.102}$$

$$F_{da_{2x}} = 2\alpha\beta \cdot \sum_{l=1}^{2} \Psi'_{l,S_{\text{aux}}} \cdot r_{l1} \cdot \mathbf{r}_l^\top \cdot \nabla(a_2^k + da_2^k) , \tag{3.103}$$

$$F_{da_{2y}} = 2\alpha\beta \cdot \sum_{l=1}^{2} \Psi'_{l,S_{\text{aux}}} \cdot r_{l2} \cdot \mathbf{r}_l^\top \cdot \nabla(a_2^k + da_2^k) , \tag{3.104}$$

$$F_{db_{1x}} = 2\alpha\beta \cdot \sum_{l=1}^{2} \Psi'_{l,S_{\text{aux}}} \cdot r_{l1} \cdot \mathbf{r}_l^\top \cdot \nabla(b_1^k + db_1^k) , \tag{3.105}$$

$$F_{db_{1y}} = 2\alpha\beta \cdot \sum_{l=1}^{2} \Psi'_{l,S_{\text{aux}}} \cdot r_{l2} \cdot \mathbf{r}_l^\top \cdot \nabla(b_1^k + db_1^k) , \tag{3.106}$$

$$F_{db_{2x}} = 2\alpha\beta \cdot \sum_{l=1}^{2} \Psi'_{l,S_{\text{aux}}} \cdot r_{l1} \cdot \mathbf{r}_l^\top \cdot \nabla(b_2^k + db_2^k) , \tag{3.107}$$

$$F_{db_{2y}} = 2\alpha\beta \cdot \sum_{l=1}^{2} \Psi'_{l,S_{\text{aux}}} \cdot r_{l2} \cdot \mathbf{r}_l^\top \cdot \nabla(b_2^k + db_2^k) \; . \tag{3.108}$$

Then, using the definitions of $\mathbf{T}_{S_2}$ and $\mathbf{T}_{S_{\text{aux}}}$, the Euler-Lagrange equations for $\mathbf{da}$ and $\mathbf{db}$ can be formulated in a vector-valued way:

$$\mathbf{0} = \mathbf{T}_{S_2}\left(\nabla(u^k + du^k) - (\mathbf{a}^k + \mathbf{da}^k)\right) + \beta \cdot \begin{pmatrix} \operatorname{div}\left(\mathbf{T}_{S_{\text{aux}}} \cdot \nabla(a_1^k + da_1^k)\right) \\ \operatorname{div}\left(\mathbf{T}_{S_{\text{aux}}} \cdot \nabla(a_2^k + da_2^k)\right) \end{pmatrix} \; , \tag{3.109}$$

$$\mathbf{0} = \mathbf{T}_{S_2}\left(\nabla(v^k + dv^k) - (\mathbf{b}^k + \mathbf{db}^k)\right) + \beta \cdot \begin{pmatrix} \operatorname{div}\left(\mathbf{T}_{S_{\text{aux}}} \cdot \nabla(b_1^k + db_1^k)\right) \\ \operatorname{div}\left(\mathbf{T}_{S_{\text{aux}}} \cdot \nabla(b_2^k + db_2^k)\right) \end{pmatrix} \; . \tag{3.110}$$

The corresponding four boundary conditions read:

$$\mathbf{n}^\top \cdot \mathbf{T}_{S_{\text{aux}}} \cdot \nabla(a_1^k + da_1^k) = 0 \; , \qquad \mathbf{n}^\top \cdot \mathbf{T}_{S_{\text{aux}}} \cdot \nabla(a_2^k + da_2^k) = 0 \; , \tag{3.111}$$

$$\mathbf{n}^\top \cdot \mathbf{T}_{S_{\text{aux}}} \cdot \nabla(b_1^k + db_1^k) = 0 \; , \qquad \mathbf{n}^\top \cdot \mathbf{T}_{S_{\text{aux}}} \cdot \nabla(b_2^k + db_2^k) \;= 0 \; . \tag{3.112}$$

Finally, when defining $\mathbf{div}$ as a vector-valued divergence operator that operates row-wise on a matrix, one can reformulate the Euler-Lagrange equations as:

$$\mathbf{0} = \mathbf{T}_{S_2}\left(\nabla(u^k + du^k) - (\mathbf{a}^k + \mathbf{da}^k)\right) + \beta \cdot \mathbf{div}\left(\mathcal{J}(\mathbf{a}^k + \mathbf{da}^k)\,\mathbf{T}_{S_{\text{aux}}}\right) \; , \tag{3.113}$$

$$\mathbf{0} = \mathbf{T}_{S_2}\left(\nabla(v^k + dv^k) - (\mathbf{b}^k + \mathbf{db}^k)\right) + \beta \cdot \mathbf{div}\left(\mathcal{J}(\mathbf{b}^k + \mathbf{db}^k)\,\mathbf{T}_{S_{\text{aux}}}\right) \; . \tag{3.114}$$

3.3 Discretization

The Euler-Lagrange equations that were derived in the previous section must be discretized to solve them numerically. As a first step towards a discretization, the lagged non-linearity method is introduced. This allows to treat all tensors $\mathbf{T}$ and all non-linear terms Ψ as constants that are independent of the unknowns.

In the following, first the discretization of two terms that appear in the Euler-Lagrange equations will be derived. Afterwards, the discretization of all Euler-Lagrange equations can be stated.

3.3.1 Discrete Anisotropic Diffusion

The first term, which appears multiple times in the necessary conditions, is the divergence of a matrix multiplied by a gradient. This is frequently used to model a diffusion process of a field u with a diffusion tensor $\mathbf{D}$. Setting it to zero corresponds to the steady state of this diffusion process, with the formula given as

$$\operatorname{div}(\mathbf{D} \cdot \nabla u) = 0 \; . \tag{3.115}$$

As Weickert [Wei98] describes, this diffusion process is based on the equilibration of concentrations as described by Fick's Law and the continuity equation. The diffusion tensor D is a positive definite symmetric matrix which due to the symmetry has the following structure:

$$\mathbf{D} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} . \tag{3.116}$$

In order to discretize Equation (3.115), a corresponding energy functional is stated

$$E(u) = \int (\nabla u)^\top \cdot \mathbf{D} \cdot \nabla u \ , \tag{3.117}$$

which has Equation (3.115) as a minimizer. To prove this, the Euler-Lagrange equation of Equation (3.117) can be calculated. If the integrand of the functional is denoted by F, the Euler-Lagrange equation reads:

$$\begin{aligned} 0 &= F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} && (3.118) \\ &= -\frac{\partial}{\partial x}(2au_x + 2bu_y) - \frac{\partial}{\partial x}(2cu_y + 2bu_x) && (3.119) \\ &= \operatorname{div}(\mathbf{D} \cdot \nabla u) \ . && (3.120) \end{aligned}$$

Thus, this functional can now be used to derive a discretization. The idea is to discretize the above energy functional and deriving it afterwards in order to then obtain a discretized divergence expression. Note that the functional differs from the corresponding part of the original energy functional of the method as described before. This is because of the lagged non-linearity approach, the diffusion tensor can here be treated as constant and independent of the unknowns.

In the following, the discretization is performed on a $M \times N$ grid with grid spacings h_x and h_y. Then, following the discretization of Weickert *et al.* [WWW13] at positions $(i + 0.5, j + 0.5)$, a discrete version of the energy functional is given by

$$E(u) = \sum_{i=0}^{N} \sum_{j=0}^{M} \left[au_x^2 + 2bu_x u_y + cu_y^2 \right]_{i+0.5,j+0.5} \ . \tag{3.121}$$

Weickert *et al.* [WWW13] further define four forward discretizations, two for u_x and two for u_y. Additional to their definition, the boundary condition is integrated using the χ indicator function:

$$\begin{aligned} [u_x]_1 &= \chi_{i+1,j} \cdot \frac{u_{i+1,j} - u_{i,j}}{h_x} \ , && (3.122) \\ [u_x]_2 &= \chi_{i+1,j+1} \cdot \frac{u_{i+1,j+1} - u_{i,j+1}}{h_x} \ , && (3.123) \\ [u_y]_1 &= \chi_{i,j+1} \cdot \frac{u_{i,j+1} - u_{i,j}}{h_y} \ , && (3.124) \\ [u_y]_2 &= \chi_{i+1,j+1} \cdot \frac{u_{i+1,j+1} - u_{i+1,j}}{h_y} \ . && (3.125) \end{aligned}$$

Then, again following [WWW13], the squared derivatives are approximated at in-between grid locations as follows:

$$\left[u_x^2\right]_{i+\frac{1}{2},j+\frac{1}{2}} \approx (1 - \alpha_{i+\frac{1}{2},j+\frac{1}{2}}) \cdot \frac{1}{2} \left([u_x]_1^2 + [u_x]_2^2 \right) + \alpha_{i+\frac{1}{2},j+\frac{1}{2}} \cdot [u_x]_1 \cdot [u_x]_2 \ , \tag{3.126}$$

$$
\begin{aligned}
\left[u_y^2\right]_{i+\frac{1}{2},j+\frac{1}{2}} &\approx \left(1-\alpha_{i+\frac{1}{2},j+\frac{1}{2}}\right)\cdot\frac{1}{2}\left([u_y]_1^2+[u_y]_2^2\right)+\alpha_{i+\frac{1}{2},j+\frac{1}{2}}\cdot[u_y]_1\cdot[u_y]_2\ , && (3.127)\\
[u_xu_y]_{i+\frac{1}{2},j+\frac{1}{2}} &\approx \frac{1-\beta_{i+\frac{1}{2},j+\frac{1}{2}}}{2}\cdot\frac{1}{2}\left([u_x]_1\cdot[u_y]_1+[u_x]_2\cdot[u_y]_2\right)\\
&\quad+\frac{1+\beta_{i+\frac{1}{2},j+\frac{1}{2}}}{2}\cdot\frac{1}{2}\left([u_x]_1\cdot[u_y]_2+[u_x]_2\cdot[u_y]_1\right)\ . && (3.128)
\end{aligned}
$$

The first two approximations are given as a linear combination of the arithmetic and geometric mean of the two approximations which can be balanced by the space-variant weight α. The weight β can also be space-variant and is used to balance the two approximations of the mixed derivative term.

The discrete energy functional can now be used to derive a space-variant stencil, similar as described in Section 2.3.3. In order to do so, the discrete energy functional is explicitly calculated for concrete values of M and N. Then it is derived w.r.t. a specific (centrally positioned) field value u_{ij} for a concrete assignment of i and j. Afterwards, the resulting term can be reorganized w.r.t. all occurring field values u_{ij}. These calculations are tedious but straightforward [WWW13], therefore the usage of a symbolic computation system such as SymPy[1] can be helpful.

The resulting space-variant stencil has the size 3×3 and determines for every position the sum of all neighborhood values. In the following, its stencil values will be denoted as $\mathcal{S}^1_{i,j,\tilde{i},\tilde{j}}$ with the pixel position (i,j) and the local offset $(\tilde{i},\tilde{j})$. Since the boundary conditions are already integrated using the χ function, it can be used directly in the implementation. For readability purposes, the full resulting stencil $\mathcal{S}^1$ has been postponed Appendix A.1.

Finally, the discretization of Equation (3.115) using the derived stencil then reads:

$$
0 = \sum_{\tilde{i},\tilde{j}\in\{-1,0,1\}}\sum \mathcal{S}^1_{i,j,\tilde{i},\tilde{j}}\cdot u_{\tilde{i},\tilde{j}} \qquad \forall i,j\ . \tag{3.129}
$$

3.3.2 Discrete Anisotropic Coupled Diffusion

A second divergence expression that needs to be discretized is the anisotropic coupled diffusion expression. It appears in the first and in the second Euler-Lagrange equation. In the following, this equation will be used together with the vector-valued coupling expression from the corresponding auxiliary Euler-Lagrange equations, which will be motivated in a moment. The two equations that shall be discretized then read

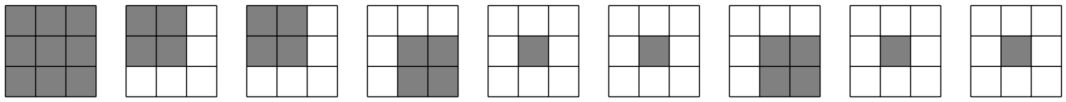
$$
\begin{aligned}
0 &= \operatorname{div}\left(\mathbf{D}\cdot(\nabla u-\mathbf{p})\right)\ , && (3.130)\\
\mathbf{0} &= \mathbf{D}\cdot(\nabla u-\mathbf{p})\ , && (3.131)
\end{aligned}
$$

where $\mathbf{D}$ is defined as in Equation (3.116) and $\mathbf{p} = (p,q)^\top$ is an arbitrary auxiliary variable.

As before, a corresponding energy functional is chosen which reads

$$
E(u) = \int (\nabla u-\mathbf{p})^\top\cdot\mathbf{D}\cdot(\nabla u-\mathbf{p})\ . \tag{3.132}
$$

[1] www.sympy.org



(a) $\mathcal{S}^{2,u,u}$ **(b)** $\mathcal{S}^{2,u,p}$ **(c)** $\mathcal{S}^{2,u,q}$ **(d)** $\mathcal{S}^{2,p,u}$ **(e)** $\mathcal{S}^{2,p,p}$ **(f)** $\mathcal{S}^{2,p,q}$ **(g)** $\mathcal{S}^{2,q,u}$ **(h)** $\mathcal{S}^{2,q,p}$ **(i)** $\mathcal{S}^{2,q,q}$

Figure 3.4: Resulting stencils for the discretization of anisotropic coupled diffusion. The result are nine stencils which in parts have globally zero entries (white) and entries that can be non-zero (gray).

To prove that the functional has Equations (3.130) and (3.131) as a minimizer, its necessary conditions are determined. Again, denoting the integrand of the functional as F, the three Euler-Lagrange equations read:

$$0 = -\frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} \Leftrightarrow 0 = \operatorname{div}\left(\mathbf{D} \cdot (\nabla u - \mathbf{p})\right) \ , \tag{3.133}$$

$$\mathbf{0} = F_{\mathbf{p}} \Leftrightarrow \mathbf{0} = \mathbf{D} \cdot (\nabla u - \mathbf{p}) \ . \tag{3.134}$$

This explains the initial choice of the two equations; considering them jointly exactly matches the energy functional above.

In the exact same manner as before, the functional is discretized. Then it is derived for a central field value and the terms are reorganized to yield a stencil. However, there are several differences to before. First, there are three possible values w.r.t. which the discretized energy functional must be derived, namely u, p and q, each corresponding to one of the Euler-Lagrange equations in Equations (3.133) and (3.134). Second, the resulting reorganization is also done w.r.t. the neighboring values of $u_{i,j}$, $p_{i,j}$ and $q_{i,j}$., so there is a stencil for each of the fields u, p and q. In total, the discretization leads to three sets of stencils, each consisting of three 3×3 stencils.

The first set of stencil corresponds to the first Euler-Lagrange equation, which derives w.r.t. u. To distinguish them from the stencil derived in the previous section, all stencils from this section are superscripted with a 2. The first three stencil are then denoted by $\mathcal{S}^{2,u,u}$, $\mathcal{S}^{2,u,p}$ and $\mathcal{S}^{2,u,q}$. The second and third set of stencils correspond to the vector-valued second Euler-Lagrange equation, which derives w.r.t. p and q. Thus, the stencils reads on the on hand $\mathcal{S}^{2,p,u}$, $\mathcal{S}^{2,p,p}$ and $\mathcal{S}^{2,p,q}$ and on the other hand $\mathcal{S}^{2,q,u}$, $\mathcal{S}^{2,q,p}$ and $\mathcal{S}^{2,q,q}$.

Note that some of the stencil values are zero. Several of the stencils are even only non-zero at the central position, so that they can be regarded more as a single coefficient rather than a stencil. However, treating them all as stencil gives a more structured representation. Figure 3.4 gives an overview of the resulting nine stencils. The entries that are zero for all spatial positions are shown in white and the potentially non-zero entries are visualized in gray.

For readability purposes, the full stencils have been postponed to Appendix A.2. Finally, the fully discretized version of Equations (3.130) and (3.131) reads

$$0 = \sum_{\tilde{i},\tilde{j}\in\{-1,0,1\}}\sum \mathcal{S}^{2,u,u}_{i,j,\tilde{i},\tilde{j}} \cdot u_{\tilde{i},\tilde{j}} + \sum_{\tilde{i},\tilde{j}\in\{-1,0,1\}}\sum \mathcal{S}^{2,u,p}_{i,j,\tilde{i},\tilde{j}} \cdot p_{\tilde{i},\tilde{j}} + \sum_{\tilde{i},\tilde{j}\in\{-1,0,1\}}\sum \mathcal{S}^{2,u,q}_{i,j,\tilde{i},\tilde{j}} \cdot q_{\tilde{i},\tilde{j}} \ , \tag{3.135}$$

$$0 = \sum_{\tilde{i},\tilde{j}\in\{-1,0,1\}}\sum \mathcal{S}^{2,p,u}_{i,j,\tilde{i},\tilde{j}} \cdot u_{\tilde{i},\tilde{j}} + \sum_{\tilde{i},\tilde{j}\in\{-1,0,1\}}\sum \mathcal{S}^{2,p,p}_{i,j,\tilde{i},\tilde{j}} \cdot p_{\tilde{i},\tilde{j}} + \sum_{\tilde{i},\tilde{j}\in\{-1,0,1\}}\sum \mathcal{S}^{2,p,q}_{i,j,\tilde{i},\tilde{j}} \cdot q_{\tilde{i},\tilde{j}} \ , \tag{3.136}$$

$$0 = \sum_{\tilde{i},\tilde{j}\in\{-1,0,1\}}\sum \mathcal{S}^{2,q,u}_{i,j,\tilde{i},\tilde{j}} \cdot u_{\tilde{i},\tilde{j}} + \sum_{\tilde{i},\tilde{j}\in\{-1,0,1\}}\sum \mathcal{S}^{2,q,p}_{i,j,\tilde{i},\tilde{j}} \cdot p_{\tilde{i},\tilde{j}} + \sum_{\tilde{i},\tilde{j}\in\{-1,0,1\}}\sum \mathcal{S}^{2,q,q}_{i,j,\tilde{i},\tilde{j}} \cdot q_{\tilde{i},\tilde{j}} \,. \tag{3.137}$$

3.3.3 Discretization of the Model

As derived in Section 3.2.2.2, the system of differential equations that has to be solved is given by

$$\begin{aligned}
0 = {} & \Psi'_{D,\mathrm{bca}} \cdot \Big(\mathbf{J}^k_{\mathrm{bca},11} \cdot du^k + \mathbf{J}^k_{\mathrm{bca},12} \cdot dv^k + \mathbf{J}^k_{\mathrm{bca},13}\Big) \\
& + \gamma \cdot \Psi'_{D,\mathrm{gca}} \cdot \Big(\mathbf{J}^k_{\mathrm{gca},11} \cdot du^k + \mathbf{J}^k_{\mathrm{gca},12} \cdot dv^k + \mathbf{J}^k_{\mathrm{gca},13}\Big) \\
& - \alpha \cdot \mathrm{div}\Big(\mathbf{T}_{S_1} \cdot \nabla(u^k + du^k)\Big) - \alpha \cdot \mathrm{div}\Big(\mathbf{T}_{S_2} \cdot \Big(\nabla(u^k + du^k) - (\mathbf{a}^k + \mathbf{da}^k)\Big)\Big) \;, \qquad (3.138) \\
0 = {} & \Psi'_{D,\mathrm{bca}} \cdot \Big(\mathbf{J}^k_{\mathrm{bca},21} \cdot du^k + \mathbf{J}^k_{\mathrm{bca},22} \cdot dv^k + \mathbf{J}^k_{\mathrm{bca},23}\Big) \\
& + \gamma \cdot \Psi'_{D,\mathrm{gca}} \cdot \Big(\mathbf{J}^k_{\mathrm{gca},21} \cdot du^k + \mathbf{J}^k_{\mathrm{gca},22} \cdot dv^k + \mathbf{J}^k_{\mathrm{gca},23}\Big) \\
& - \alpha \cdot \mathrm{div}\Big(\mathbf{T}_{S_1} \cdot \nabla(v^k + dv^k)\Big) - \alpha \cdot \mathrm{div}\Big(\mathbf{T}_{S_2} \cdot \Big(\nabla(v^k + dv^k) - (\mathbf{b}^k + \mathbf{db}^k)\Big)\Big) \;, \qquad (3.139) \\
\mathbf{0} = {} & \mathbf{T}_{S_2}\Big(\nabla(u^k + du^k) - (\mathbf{a}^k + \mathbf{da}^k)\Big) + \beta \cdot \mathbf{div}\Big(\mathcal{J}(\mathbf{a}^k + \mathbf{da}^k)\,\mathbf{T}_{S_{\mathrm{aux}}}\Big) \;, \qquad (3.140) \\
\mathbf{0} = {} & \mathbf{T}_{S_2}\Big(\nabla(v^k + dv^k) - (\mathbf{b}^k + \mathbf{db}^k)\Big) + \beta \cdot \mathbf{div}\Big(\mathcal{J}(\mathbf{b}^k + \mathbf{db}^k)\,\mathbf{T}_{S_{\mathrm{aux}}}\Big) \;. \qquad (3.141)
\end{aligned}$$

In order to discretize the Euler-Lagrange equations, the previously derived stencils $\mathcal{S}^1$ and $\mathcal{S}^2$ are used several times. Note that $\mathcal{S}^1$ can easily repurposed to work on any other field. In the following, it will be used for the diffusion of $u^k + du^k$ and $v^k + dv^k$ with tensor $\mathbf{T}_{S_1}$ and additionally for the diffusion of all four auxiliary variables with $\mathbf{T}_{S_{\mathrm{aux}}}$. Similarly, the stencil $\mathcal{S}^2$ can be used for any field variable with any coupling variable. Thus, it will be used in the following for a coupled diffusion of $u^k + du^k$ with $\mathbf{a}^k + \mathbf{da}^k$ and $u^k + dv^k$ with $\mathbf{b}^k + \mathbf{db}^k$.

Finally, the completely discretized Euler-Lagrange equations of the described model then read:

$$\begin{aligned}
0 = {} & [\Psi'_{D,\mathrm{bca}}]_{i,j} \cdot \Big([\mathbf{J}^k_{\mathrm{bca},11}]_{i,j} \cdot du^k_{i,j} + [\mathbf{J}^k_{\mathrm{bca},12}]_{i,j} \cdot dv^k_{i,j} + [\mathbf{J}^k_{\mathrm{bca},13}]_{i,j}\Big) \\
& + \gamma \cdot [\Psi'_{D,\mathrm{gca}}]_{i,j} \cdot \Big([\mathbf{J}^k_{\mathrm{gca},11}]_{i,j} \cdot du^k_{i,j} + [\mathbf{J}^k_{\mathrm{gca},12}]_{i,j} \cdot dv^k_{i,j} + [\mathbf{J}^k_{\mathrm{gca},13}]_{i,j}\Big) \\
& - \alpha \cdot \sum_{\tilde{i},\tilde{j}\in\{-1,0,1\}}\sum \mathcal{S}^1_{i,j,\tilde{i},\tilde{j}} \cdot (u^k_{\tilde{i},\tilde{j}} + du^k_{\tilde{i},\tilde{j}}) \\
& - \alpha \cdot \sum_{\tilde{i},\tilde{j}\in\{-1,0,1\}}\sum \mathcal{S}^{2,u,u}_{i,j,\tilde{i},\tilde{j}} \cdot (u^k_{\tilde{i},\tilde{j}} + du^k_{\tilde{i},\tilde{j}}) + \mathcal{S}^{2,u,a_1}_{i,j,\tilde{i},\tilde{j}} \cdot (a^k_{1,\tilde{i},\tilde{j}} + da^k_{1,\tilde{i},\tilde{j}}) + \mathcal{S}^{2,u,a_2}_{i,j,\tilde{i},\tilde{j}} \cdot (a^k_{2,\tilde{i},\tilde{j}} + da^k_{2,\tilde{i},\tilde{j}}) \;,
\end{aligned} \tag{3.142}$$

$$
\begin{aligned}
0 = {} & [\Psi'_{D,\mathrm{bca}}]_{i,j} \cdot \Big([\mathbf{J}^k_{\mathrm{bca},21}]_{i,j} \cdot du^k_{i,j} + [\mathbf{J}^k_{\mathrm{bca},22}]_{i,j} \cdot dv^k_{i,j} + [\mathbf{J}^k_{\mathrm{bca},23}]_{i,j} \Big) \\
& + \gamma \cdot [\Psi'_{D,\mathrm{gca}}]_{i,j} \cdot \Big([\mathbf{J}^k_{\mathrm{gca},21}]_{i,j} \cdot du^k_{i,j} + [\mathbf{J}^k_{\mathrm{gca},22}]_{i,j} \cdot dv^k_{i,j} + [\mathbf{J}^k_{\mathrm{gca},23}]_{i,j} \Big) \\
& - \alpha \cdot \sum\sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \mathcal{S}^{1}_{i,j,\tilde{i},\tilde{j}} \cdot (v^k_{\tilde{i},\tilde{j}} + dv^k_{\tilde{i},\tilde{j}}) \\
& - \alpha \cdot \sum\sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \mathcal{S}^{2,v,v}_{i,j,\tilde{i},\tilde{j}} \cdot (v^k_{\tilde{i},\tilde{j}} + dv^k_{\tilde{i},\tilde{j}}) + \mathcal{S}^{2,v,b_1}_{i,j,\tilde{i},\tilde{j}} \cdot (b^k_{1,\tilde{i},\tilde{j}} + db^k_{1,\tilde{i},\tilde{j}}) + \mathcal{S}^{2,v,b_2}_{i,j,\tilde{i},\tilde{j}} \cdot (b^k_{2,\tilde{i},\tilde{j}} + db^k_{2,\tilde{i},\tilde{j}}) \;,
\end{aligned}
\tag{3.143}
$$

$$
\begin{aligned}
0 = {} & \sum\sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \mathcal{S}^{2,a_1,u}_{i,j,\tilde{i},\tilde{j}} \cdot (u^k_{\tilde{i},\tilde{j}} + du^k_{\tilde{i},\tilde{j}}) + \mathcal{S}^{2,a_1,a_1}_{i,j,\tilde{i},\tilde{j}} \cdot (a^k_{1,\tilde{i},\tilde{j}} + da^k_{1,\tilde{i},\tilde{j}}) + \mathcal{S}^{2,a_1,a_2}_{i,j,\tilde{i},\tilde{j}} \cdot (a^k_{2,\tilde{i},\tilde{j}} + da^k_{2,\tilde{i},\tilde{j}}) \\
& + \beta \cdot \sum\sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \mathcal{S}^{1}_{i,j,\tilde{i},\tilde{j}} \cdot (a^k_{1,\tilde{i},\tilde{j}} + da^k_{1,\tilde{i},\tilde{j}}) \;,
\end{aligned}
\tag{3.144}
$$

$$
\begin{aligned}
0 = {} & \sum\sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \mathcal{S}^{2,a_2,u}_{i,j,\tilde{i},\tilde{j}} \cdot (u^k_{\tilde{i},\tilde{j}} + du^k_{\tilde{i},\tilde{j}}) + \mathcal{S}^{2,a_2,a_1}_{i,j,\tilde{i},\tilde{j}} \cdot (a^k_{1,\tilde{i},\tilde{j}} + da^k_{1,\tilde{i},\tilde{j}}) + \mathcal{S}^{2,a_2,a_2}_{i,j,\tilde{i},\tilde{j}} \cdot (a^k_{2,\tilde{i},\tilde{j}} + da^k_{2,\tilde{i},\tilde{j}}) \\
& + \beta \cdot \sum\sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \mathcal{S}^{1}_{i,j,\tilde{i},\tilde{j}} \cdot (a^k_{2,\tilde{i},\tilde{j}} + da^k_{2,\tilde{i},\tilde{j}}) \;,
\end{aligned}
\tag{3.145}
$$

$$
\begin{aligned}
0 = {} & \sum\sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \mathcal{S}^{2,b_1,v}_{i,j,\tilde{i},\tilde{j}} \cdot (v^k_{\tilde{i},\tilde{j}} + dv^k_{\tilde{i},\tilde{j}}) + \mathcal{S}^{2,b_1,b_1}_{i,j,\tilde{i},\tilde{j}} \cdot (b^k_{1,\tilde{i},\tilde{j}} + db^k_{1,\tilde{i},\tilde{j}}) + \mathcal{S}^{2,b_1,b_2}_{i,j,\tilde{i},\tilde{j}} \cdot (b^k_{2,\tilde{i},\tilde{j}} + db^k_{2,\tilde{i},\tilde{j}}) \\
& + \beta \cdot \sum\sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \mathcal{S}^{1}_{i,j,\tilde{i},\tilde{j}} \cdot (b^k_{1,\tilde{i},\tilde{j}} + db^k_{1,\tilde{i},\tilde{j}}) \;,
\end{aligned}
\tag{3.146}
$$

$$
\begin{aligned}
0 = {} & \sum\sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \mathcal{S}^{2,b_2,v}_{i,j,\tilde{i},\tilde{j}} \cdot (v^k_{\tilde{i},\tilde{j}} + dv^k_{\tilde{i},\tilde{j}}) + \mathcal{S}^{2,b_2,b_1}_{i,j,\tilde{i},\tilde{j}} \cdot (b^k_{1,\tilde{i},\tilde{j}} + db^k_{1,\tilde{i},\tilde{j}}) + \mathcal{S}^{2,b_2,b_2}_{i,j,\tilde{i},\tilde{j}} \cdot (b^k_{2,\tilde{i},\tilde{j}} + db^k_{2,\tilde{i},\tilde{j}}) \\
& + \beta \cdot \sum\sum_{\tilde{i},\tilde{j} \in \{-1,0,1\}} \mathcal{S}^{1}_{i,j,\tilde{i},\tilde{j}} \cdot (b^k_{2,\tilde{i},\tilde{j}} + db^k_{2,\tilde{i},\tilde{j}}) \;.
\end{aligned}
\tag{3.147}
$$

3.4 Structure of the Linear System of Equations

The discrete Euler-Lagrange equations that were derived in the last section form a linear system of equations. Since there are six Euler-Lagrange equations, the total number of equations is $6 \cdot M \cdot N$ for an image of size $M \times N$.

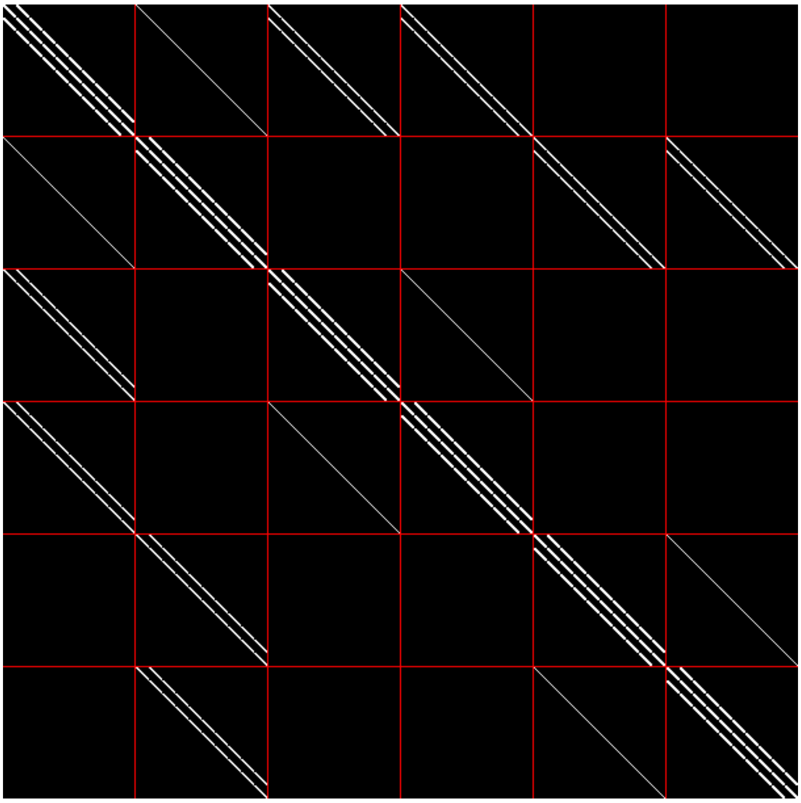


Figure 3.5: Visualization of Matrix $\mathbf{A}$ including coupling variables for an image size of 10×10. Non-zero entries are shown in white, while zero entries are visualized in black. To better view the block structure, red helper lines were inserted. Best viewed digitally.

This system of equation again has the form $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$, where $\mathbf{x}$ consists of all unknowns, i.e. the flow entries and all auxiliary variables:

$$\mathbf{x} = \begin{pmatrix} u_{i,j} \\ \hline v_{i,j} \\ \hline a_{1,i,j} \\ \hline a_{2,i,j} \\ \hline b_{1,i,j} \\ \hline b_{2,i,j} \end{pmatrix} . \tag{3.148}$$

The vector $\mathbf{b}$ contains all terms not dependent on any of the unknowns. Finally, matrix $\mathbf{A}$ has size $(6MN) \times (6MN)$ and models all dependencies between unknowns.

Figure 3.5 shows a visualization of the matrix where non-zero entries are visualized as a white pixel and the zero entries in black. Additionally, red helper lines are inserted to show the 6×6 block structure the matrix has. Since the unknowns are stacked as described above, one can read from the different sections which variables are coupled. The top-left 2×2 blocks represent the coupling of u and v and have a strong similarity to the matrix described and visualized in Section 2.3.4. The other parts represent the coupling of the auxiliary variables with the flow variables and with each other.

4 Order-Adaptive Regularization with an Anisotropic Selection Scheme

So far, the presented approach strives to select the optimal regularization order for every pixel location. One shortcoming of this is that for each local image direction, the same regularization order is chosen. In the following, an extension to the model of Maurer *et al.* [MSB17b] will be presented that individually selects the optimal regularization order for each local image direction. The idea is to allow even more adaptation to local structures by introducing the possibility of locally selecting the order individually for each direction. As for previous anisotropic approaches, two directions are introduced that relate to the local image structure, with one pointing in the dominant direction and the other orthogonal to it. As a result, the novel method estimates not only a single order selection parameter, but a set of two parameters, corresponding to the two directions.

4.1 Model

In the following, the novel model extending the method of Maurer *et al.* [MSB17b] to include an anisotropic order selection will be introduced. To this end, an energy functional will be introduced, where first the data term and afterwards the smoothness term will be discussed.

4.1.1 Data Term

Since the main focus of this thesis lies on the regularization term, the data term is chosen the same as before. The data term includes a brightness and gradient constancy assumption without linearization, which are robustified using a sub-quadratic penalizer. It is given in Equation (3.2) and reads

$$D(\mathbf{u}) = \int_\Omega \Psi_D \left((f(\mathbf{x}+\mathbf{u}, t+1) - f(\mathbf{x}, t))^2 \right) + \gamma \cdot \Psi_D \left(\|\nabla f(\mathbf{x}+\mathbf{u}, t+1) - \nabla f(\mathbf{x}, t)\|_2^2 \right) \, d\mathbf{x} \,. \tag{4.1}$$

4.1.2 Smoothness Term

The smoothness term introduces the anisotropic order selection method, where the goal is to not only allow for a local order adaptation, but to additionally allow adaptive behavior in each local direction. Similar to before, a first- and a second-order anisotropic regularization term are integrated in the smoothness term. But while in the previous approach a comparison between the whole first- and second-order term yielded the optimal regularization order, this comparison is now done for each directional component of the terms individually. This allows to select a potentially different order for each direction independently, which results in introducing two order-selection parameters. In the approach by Maurer *et al.* [MSB17b], the order selection parameter was used for a convex combination of the complete first- and second-order terms. But, since both of these terms are already given anisotropic, now each of their directional components is treated individually. This results in two convex combinations relating to the two directions that combine for each term the component relating to the direction.

In order to realize this, some formulations have to be adapted. First, instead of the scalar order selection parameter c introduced in Section 3.1.2, now a vector-valued parameter $\mathbf{c}$ is used:

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} , \tag{4.2}$$

where c_1 corresponds to the selected order in $\mathbf{r}_1$-direction and c_2 to the order in $\mathbf{r}_2$-direction.

As stated before, in the following, the anisotropic directional components of the first- and second-order smoothness terms as introduced in Section 3.1.2 will be treated individually. Thus, in order to allow for a formulation per direction, the first- and second-order regularization terms S_1 and S_2 are split up into two components S_1^l and S_2^l for $l \in \{1, 2\}$ which are related to the two corresponding $\mathbf{r}_l$-directions:

$$S_1(\mathbf{u}) = \sum_{l=1}^{2} \Psi_l\Big((\mathbf{r}_l^\top \nabla u)^2 + (\mathbf{r}_l^\top \nabla v)^2\Big) = \sum_{l=1}^{2} S_1^l(\mathbf{u}) \tag{4.3}$$

with

$$S_1^l(\mathbf{u}) = \Psi_l\Big((\mathbf{r}_l^\top \nabla u)^2 + (\mathbf{r}_l^\top \nabla v)^2\Big) , \tag{4.4}$$

and

$$S_2(\mathbf{u}, \mathbf{a}, \mathbf{b}) = \sum_{l=1}^{2} \Psi_l\Big((\mathbf{r}_l^\top (\nabla u - \mathbf{a}))^2 + (\mathbf{r}_l^\top (\nabla v - \mathbf{b}))^2\Big) = \sum_{l=1}^{2} S_2^l(\mathbf{u}, \mathbf{a}, \mathbf{b}) \tag{4.5}$$

with

$$S_2^l(\mathbf{u}, \mathbf{a}, \mathbf{b}) = \Psi_l\Big((\mathbf{r}_l^\top (\nabla u - \mathbf{a}))^2 + (\mathbf{r}_l^\top (\nabla v - \mathbf{b}))^2\Big) . \tag{4.6}$$

Now all prerequisites are introduced and the regularization term can be presented. As described, the first- and second order terms are combined as a convex combination, with the difference of now doing this for each direction individually. Then the base model is given by the functional

$$R(\mathbf{u}, \mathbf{c}) = \int_\Omega \inf_{\mathbf{a},\mathbf{b}} \Big\{ \sum_{l=1}^{2} c_l \cdot S_1^l(\mathbf{u}) + \sum_{l=1}^{2} (1 - c_l) \cdot S_2^l(\mathbf{u}, \mathbf{a}, \mathbf{b}) + \beta \cdot S_{\text{aux}}(\mathbf{a}, \mathbf{b}) + \sum_{l=1}^{2} \phi_{l\lambda}(c_l) \Big\} \, d\mathbf{x} \, . \tag{4.7}$$

In contrast to the isotropic selection scheme of Maurer *et al.* [MSB17b] introduced in the previous chapter, now the parameter c_l is used as a multiplier per l-direction. Also, two selection terms $\phi_{1\lambda}(c_1)$ and $\phi_{2\lambda}(c_2)$ are used instead of one.

Derivation of Selection Terms In the following, the two selection terms will be derived, analogous to the derivation in Section 3.1.2. Again the goal is to have a first- and a second-order term competing against each other, where the superior one decides on the resulting regularization order. In order to realize this, similar to method of Maurer *et al.* [MSB17b], the difference of two energies is integrated, a threshold is added and the sigmoid function is applied to the result. Then it is straightforward that c_1 and c_2 are chosen as follows:

$$c_1 = \frac{1}{1+e^{-\Delta_1/\lambda}} \quad \text{with} \quad \Delta_1 = T_1 + \frac{1}{|\Omega|} \cdot \int_\Omega S_2^1 - S_1^1 \, d\mathbf{x} \; , \tag{4.8}$$

$$c_2 = \frac{1}{1+e^{-\Delta_2/\lambda}} \quad \text{with} \quad \Delta_2 = T_2 + \frac{1}{|\Omega|} \cdot \int_\Omega S_2^2 - S_1^2 \, d\mathbf{x} \; . \tag{4.9}$$

Note that in contrast to before, the energy difference of S_1 and S_2 is now calculated only from the component relating to its respective direction: $S_2^l - S_1^l$ for $l \in \{1, 2\}$. Also, a threshold T_l is introduced for each direction individually. The reason for introducing two thresholds is that it is possible to select different penalizer functions Ψ_l for each direction. Thus, the magnitude of the values that the threshold is applied to might be different, so a threshold for each component individually makes sense.

Similar to the previous chapter, the goal is to derive two selection terms that model exactly the properties of c_1 and c_2 as described in Equations (4.8) and (4.9). In order to do so, both equations can now be simplified following the exact same steps as in Equations (3.9) to (3.11). The result reads

$$\int_\Omega S_1^1 - S_2^1 \, d\mathbf{x} = |\Omega| \left(\lambda \cdot \ln \left(\frac{1}{c_1} - 1 \right) + T_1 \right) \; , \tag{4.10}$$

$$\int_\Omega S_1^2 - S_2^2 \, d\mathbf{x} = |\Omega| \left(\lambda \cdot \ln \left(\frac{1}{c_2} - 1 \right) + T_2 \right) \; . \tag{4.11}$$

As before, from stating that the optimal c_1 and c_2 should be a minimizer of R, it follows that the selection terms $\phi_{1\lambda}$ and $\phi_{2\lambda}$ must read

$$\phi_{1\lambda}(c_1) = \lambda \cdot \phi(c_1) + (1 - c_1) \cdot T_1 \; , \tag{4.12}$$

$$\phi_{2\lambda}(c_2) = \lambda \cdot \phi(c_2) + (1 - c_2) \cdot T_2 \; , \tag{4.13}$$

using again the definition

$$\phi(c_l) := c_l \cdot \ln(c_l) + (1 - c_l) \cdot \ln(1 - c_l) \; . \tag{4.14}$$

These derivation steps again follow the steps in Equations (3.12) to (3.20) where they are given in detail. Finally, the derived selection terms can be inserted into the regularization term. The complete regularization term based on an anisotropic order selection mechanism then reads

$$R(\mathbf{u}, \mathbf{c}) = \int_\Omega \inf_{\mathbf{a},\mathbf{b}} \Big\{ \sum_{l=1}^{2} c_l \cdot S_1^l(\mathbf{u}) + (1 - c_l) \cdot \Big(S_2^l(\mathbf{u}, \mathbf{a}, \mathbf{b}) + T_l \Big) \\ + \beta \cdot S_{\text{aux}}(\mathbf{a}, \mathbf{b}) + \lambda \cdot \phi(c_l) \Big\} \, d\mathbf{x} \; . \tag{4.15}$$

Non-Local Anisotropic Order Selection In the same fashion as before, $\mathbf{c}$ can be chosen as a spatially variant function $\mathbf{c}(\mathbf{x})$, which is embedded in a non-local scheme. Again, this leads to integrating $\mathbf{c}$ over a local neighborhood $\mathcal{N}$. The non-local regularization term then reads:

$$R(\mathbf{u},\mathbf{c}) = \int_\Omega \inf_{\mathbf{a},\mathbf{b}} \Big\{ \sum_{l=1}^{2} \bar{c}_l \cdot S_1^l(\mathbf{u}) + \sum_{l=1}^{2} (1-\bar{c}_l) \cdot \Big(S_2^l(\mathbf{u},\mathbf{a},\mathbf{b}) + T_l \Big) \\ + \beta \cdot S_{\text{aux}}(\mathbf{a},\mathbf{b}) + \sum_{l=1}^{2} \lambda \cdot \phi(c_l) \Big\} \, d\mathbf{x} \; , \tag{4.16}$$

where

$$\bar{c}_l(\mathbf{x}) = \frac{1}{|\mathcal{N}(\mathbf{x})|} \int_{\mathcal{N}(\mathbf{x})} c_l(\mathbf{y}) \, d\mathbf{y} \; . \tag{4.17}$$

As defined previously, $\mathbf{c}(\mathbf{x})$ is then calculated for each dimension l as

$$c_l(\mathbf{x}) = \frac{1}{1+e^{-\Delta_l/\lambda}} \; , \tag{4.18}$$

with

$$\Delta_l = \int_{\mathcal{N}(\mathbf{x})} \frac{1}{|\mathcal{N}(\mathbf{y})|} \Big(T_l \; + \; S_2^l\big(\mathbf{u}(\mathbf{y}),\mathbf{a}(\mathbf{y}),\mathbf{b}(\mathbf{y})\big) \; - \; S_1^l\big(\mathbf{u}(\mathbf{y})\big) \Big) \, d\mathbf{y} \; . \tag{4.19}$$

As in the previous chapter, this non-local formulation will be used in the following.

4.2 Minimization

The derivation of the minimization follows the same structure as in Section 3.2. First, a differential formulation is given, which relates to the warping scheme. Second, the alternating minimization is presented, which alternates between estimating the order selection parameters and the flow. To this end, the necessary condition to calculate the order selection parameters and the minimality conditions of the flow estimation are given.

4.2.1 Differential Formulation

Data Term The introduction of anisotropic order selection has no influence on the data term, thus the differential formulation of the data term can be found in Equation (3.38). As before, the extended flow definition is used, which reads

$$\mathbf{dw}^k = \Big((\mathbf{du}^k)^\top, 1 \Big)^\top = \Big(du^k, dv^k, 1 \Big)^\top \; . \tag{4.20}$$

Then, the differential data term reads

$$D(\mathbf{du}^k) = \int_\Omega \Psi_D \left((\mathbf{dw}^k)^\top \mathbf{J}_{\text{bca}}^k \, \mathbf{dw}^k \right) + \gamma \cdot \Psi_D \left((\mathbf{dw}^k)^\top \mathbf{J}_{\text{gca}}^k \, \mathbf{dw}^k \right) \; d\mathbf{x} \; . \tag{4.21}$$

Regularization Term The differential formulation of the regularization term with anisotropic order selection bears only small changes to the isotropic case. The regularization term reads

$$\begin{aligned} R(\mathbf{du}^k, \mathbf{c}^k) = \int_\Omega \inf_{\mathbf{da}^k, \mathbf{db}^k} \Big\{ & \sum_{l=1}^{2} \bar{c}_l^k(\mathbf{x}) \cdot S_1^l(\mathbf{u}^k + \mathbf{du}^k) \\ & + \sum_{l=1}^{2} (1 - \bar{c}_l^k(\mathbf{x})) \cdot \Big(S_2^l(\mathbf{u}^k + \mathbf{du}^k, \mathbf{a}^k + \mathbf{da}^k, \mathbf{b}^k + \mathbf{db}^k) + T_l \Big) \\ & + \beta \cdot S_{\text{aux}}(\mathbf{a}^k + \mathbf{da}^k, \mathbf{b}^k + \mathbf{db}^k) \\ & + \sum_{l=1}^{2} \lambda \cdot \phi(c_l^k(\mathbf{x})) \Big\} \, d\mathbf{x} \;, \end{aligned} \tag{4.22}$$

with

$$S_1^l(\mathbf{u}^k + \mathbf{du}^k) = \Psi_l\Big((\mathbf{r}_l^\top \nabla(u^k + du^k))^2 + (\mathbf{r}_l^\top \nabla(v^k + dv^k))^2 \Big) \;, \tag{4.23}$$

$$\begin{aligned} S_2^l(\mathbf{u}^k + \mathbf{du}^k, \mathbf{a}^k + \mathbf{da}^k, \mathbf{b}^k + \mathbf{db}^k) = \Psi_l \Bigg(& \Big(\mathbf{r}_l^\top \Big(\nabla(u^k + du^k) - (\mathbf{a}^k + \mathbf{da}^k) \Big) \Big)^2 \\ & + \Big(\mathbf{r}_l^\top \Big(\nabla(v^k + dv^k) - (\mathbf{b}^k + \mathbf{db}^k) \Big) \Big)^2 \Bigg) \;, \end{aligned} \tag{4.24}$$

$$\begin{aligned} S_{\text{aux}}(\mathbf{a}^k + \mathbf{da}^k, \mathbf{b}^k + \mathbf{db}^k) = \sum_{l=1}^{2} \Psi_l \Bigg(\sum_{k=1}^{2} & \Big(\mathbf{r}_k^\top \mathcal{J}(\mathbf{a}^k + \mathbf{da}^k) \, \mathbf{r}_l \Big)^2 \\ & + \Big(\mathbf{r}_k^\top \mathcal{J}(\mathbf{b}^k + \mathbf{db}^k) \, \mathbf{r}_l \Big)^2 \Bigg) \;. \end{aligned} \tag{4.25}$$

4.2.2 Alternating Minimization

Analogously to before, an alternation between computing $\mathbf{c}^{k+1}$ given $\mathbf{du}^k, \mathbf{da}^k, \mathbf{db}^k$ and computing $\mathbf{du}^{k+1}, \mathbf{da}^{k+1}, \mathbf{db}^{k+1}$ given $\mathbf{c}^{k+1}$ is introduced:

$$\mathbf{c}^{k+1} = \arg\min_{\mathbf{c}} E(\mathbf{du}^k, \mathbf{c}) \;, \tag{4.26}$$

$$\mathbf{du}^{k+1}, \mathbf{da}^{k+1}, \mathbf{db}^{k+1} = \arg\min_{\mathbf{du}, \mathbf{da}, \mathbf{db}} E(\mathbf{du}, \mathbf{c}^{k+1}) \;. \tag{4.27}$$

Note that although both minimization steps are formulated in a vector-valued manner, the minimizations w.r.t. the order maps c_1 and c_2 are independent of each other, while this is not the case for the minimization w.r.t. the flow.

First, the minimization w.r.t. the order selection maps $\mathbf{c}$ will be discussed, followed by the minimization w.r.t. the flow.

4.2.2.1 Minimization w.r.t. the Order Selection Maps

The first alternating step is the minimization w.r.t. the order selection maps, where the flow and auxiliary variables are kept fixed. Similar as before, when the energy is minimized w.r.t. $\mathbf{c}$, all terms not containing $\mathbf{c}$ can be omitted. The remaining energy functional $E_{\mathbf{c}}$ is given by

$$\begin{aligned} E_{\mathbf{c}}(\mathbf{c}) = \alpha \cdot \int_{\Omega} \inf_{\mathbf{da}^k,\mathbf{db}^k} \Big\{ & \sum_{l=1}^{2} \bar{c}_l \cdot S_1^l(\mathbf{u}^k + \mathbf{du}^k) \\ & + \sum_{l=1}^{2} (1-\bar{c}_l) \cdot \Big(S_2^l(\mathbf{u}^k + \mathbf{du}^k, \mathbf{a}^k + \mathbf{da}^k, \mathbf{b}^k + \mathbf{db}^k) + T_l \Big) \\ & + \sum_{l=1}^{2} \lambda \cdot \phi(c_l) \Big\} \, d\mathbf{x} \ . \end{aligned} \tag{4.28}$$

Using the same reformulation strategy as described in Section 3.2.2, the regularization term can be defined pointwise for all $\mathbf{c}$:

$$\begin{aligned} E_{\mathbf{c}}(\mathbf{c}) = \alpha \cdot \int_{\Omega} \inf_{\mathbf{da}^k,\mathbf{db}^k} \Big\{ & \sum_{l=1}^{2} c_l(\mathbf{x}) \cdot \left(\int_{\mathcal{N}(\mathbf{x})} \frac{1}{|\mathcal{N}(\mathbf{y})|} \, S_1^l \, d\mathbf{y} \right) \\ & + \sum_{l=1}^{2} S_2^l + T_l - c_l(\mathbf{x}) \cdot \left(\int_{\mathcal{N}(\mathbf{x})} \frac{1}{|\mathcal{N}(\mathbf{y})|} \Big(S_2^l + T_l \Big) \, d\mathbf{y} \right) \\ & + \sum_{l=1}^{2} \lambda \cdot \phi(c_l) \Big\} \, d\mathbf{x} \ . \end{aligned} \tag{4.29}$$

Now, all occurrences of $\mathbf{c}$ are defined pointwise and the functional has the form

$$E_{\mathbf{c}}(\mathbf{c}) = \int_{\Omega} F(\mathbf{x}, \mathbf{c}) \, d\mathbf{x} \ . \tag{4.30}$$

Since no derivatives of $\mathbf{c}$ are present, the resulting two Euler-Lagrange equations are given by

$$F_{c_1} = 0 \ , \tag{4.31}$$

$$F_{c_2} = 0 \ . \tag{4.32}$$

Similar to before, no additional boundary conditions must hold. Analogously to the steps in Section 3.2.2, the equations come down to the previous definitions of $\mathbf{c}$ in Equations (4.8) and (4.9):

$$c_1 = \frac{1}{1 + e^{-\Delta_1/\lambda}} \ , \tag{4.33}$$

$$c_2 = \frac{1}{1 + e^{-\Delta_2/\lambda}} \ , \tag{4.34}$$

with

$$\Delta_1 = \int_{\mathcal{N}(\mathbf{x})} \frac{1}{|\mathcal{N}(\mathbf{y})|} \cdot \big(T_1 + S_2^1 - S_1^1 \big) \ , \tag{4.35}$$

$$\Delta_2 = \int_{\mathcal{N}(\mathbf{x})} \frac{1}{|\mathcal{N}(\mathbf{y})|} \cdot \big(T_2 + S_2^2 - S_1^2 \big) \ . \tag{4.36}$$

Both Euler-Lagrange equations can be solved in closed form and analytically.

4.2.2.2 Minimization w.r.t. the Flow

As a second alternating step, the minimization w.r.t. the flow and auxiliary variables is done. In order to achieve this, the necessary conditions of the energy functional are stated, which read

$$\begin{aligned}
0 &= \Psi'_{D,\mathrm{bca}} \cdot \left(\mathbf{J}^k_{\mathrm{bca},11} \cdot du^k + \mathbf{J}^k_{\mathrm{bca},12} \cdot dv^k + \mathbf{J}^k_{\mathrm{bca},13}\right) \\
&\quad + \gamma \cdot \Psi'_{D,\mathrm{gca}} \cdot \left(\mathbf{J}^k_{\mathrm{gca},11} \cdot du^k + \mathbf{J}^k_{\mathrm{gca},12} \cdot dv^k + \mathbf{J}^k_{\mathrm{gca},13}\right) \\
&\quad - \alpha \cdot \mathrm{div}\left(\mathbf{T}_{S_1} \cdot \nabla(u^k + du^k)\right) - \alpha \cdot \mathrm{div}\left(\mathbf{T}_{S_2} \cdot \left(\nabla(u^k + du^k) - (\mathbf{a}^k + \mathbf{da}^k)\right)\right) \ , \qquad (4.37) \\
0 &= \Psi'_{D,\mathrm{bca}} \cdot \left(\mathbf{J}^k_{\mathrm{bca},21} \cdot du^k + \mathbf{J}^k_{\mathrm{bca},22} \cdot dv^k + \mathbf{J}^k_{\mathrm{bca},23}\right) \\
&\quad + \gamma \cdot \Psi'_{D,\mathrm{gca}} \cdot \left(\mathbf{J}^k_{\mathrm{gca},21} \cdot du^k + \mathbf{J}^k_{\mathrm{gca},22} \cdot dv^k + \mathbf{J}^k_{\mathrm{gca},23}\right) \\
&\quad - \alpha \cdot \mathrm{div}\left(\mathbf{T}_{S_1} \cdot \nabla(v^k + dv^k)\right) - \alpha \cdot \mathrm{div}\left(\mathbf{T}_{S_2} \cdot \left(\nabla(v^k + dv^k) - (\mathbf{b}^k + \mathbf{db}^k)\right)\right) \ , \qquad (4.38) \\
\mathbf{0} &= \mathbf{T}_{S_2}\left(\nabla(u^k + du^k) - (\mathbf{a}^k + \mathbf{da}^k)\right) + \beta \cdot \mathbf{div}\left(\mathcal{J}(\mathbf{a}^k + \mathbf{da}^k)\,\mathbf{T}_{S_\mathrm{aux}}\right) \ , \qquad (4.39) \\
\mathbf{0} &= \mathbf{T}_{S_2}\left(\nabla(v^k + dv^k) - (\mathbf{b}^k + \mathbf{db}^k)\right) + \beta \cdot \mathbf{div}\left(\mathcal{J}(\mathbf{b}^k + \mathbf{db}^k)\,\mathbf{T}_{S_\mathrm{aux}}\right) \ , \qquad (4.40)
\end{aligned}$$

with the following boundary conditions:

$$\begin{aligned}
&\mathbf{n}^\top \cdot \left(\mathbf{T}_{S_1} \cdot \nabla(u^k + du^k) + \mathbf{T}_{S_2} \cdot \left(\nabla(u^k + du^k) - (\mathbf{a}^k + \mathbf{da}^k)\right)\right) = 0 \ , \qquad (4.41) \\
&\mathbf{n}^\top \cdot \left(\mathbf{T}_{S_1} \cdot \nabla(v^k + dv^k) + \mathbf{T}_{S_2} \cdot \left(\nabla(v^k + dv^k) - (\mathbf{b}^k + \mathbf{db}^k)\right)\right) = 0 \ , \qquad (4.42) \\
&\mathbf{n}^\top \cdot \mathbf{T}_{S_\mathrm{aux}} \cdot \nabla(a_1^k + da_1^k) = 0 \ , \qquad (4.43) \\
&\mathbf{n}^\top \cdot \mathbf{T}_{S_\mathrm{aux}} \cdot \nabla(a_2^k + da_2^k) = 0 \ , \qquad (4.44) \\
&\mathbf{n}^\top \cdot \mathbf{T}_{S_\mathrm{aux}} \cdot \nabla(b_1^k + db_1^k) = 0 \ , \qquad (4.45) \\
&\mathbf{n}^\top \cdot \mathbf{T}_{S_\mathrm{aux}} \cdot \nabla(b_2^k + db_2^k) = 0 \ . \qquad (4.46)
\end{aligned}$$

Since all occurrences of **c** are wrapped inside the diffusion tensors $\mathbf{T}_{S_1}$ and $\mathbf{T}_{S_2}$, the Euler-Lagrange equations have the exact same structure as before. The only difference, however, lies in the definitions of the diffusion tensors $\mathbf{T}_{S_1}$ and $\mathbf{T}_{S_2}$:

$$\mathbf{T}_{S_1} = \sum_{l=1}^{2} \bar{c}_l \cdot \Psi'_{l,S_1} \cdot \mathbf{r}_l \cdot \mathbf{r}_l^\top \ , \qquad (4.47)$$

$$\mathbf{T}_{S_2} = \sum_{l=1}^{2} (1 - \bar{c}_l) \cdot \Psi'_{l,S_2} \cdot \mathbf{r}_l \cdot \mathbf{r}_l^\top \ . \qquad (4.48)$$

When comparing these tensors to their previous definition in Section 3.2.2.2, the difference lies in the fact that $\bar{c}_l$ is dependent on the direction l. Thus, it becomes obvious that for $c_1 = c_2$, the diffusion tensors come down to their previous definition. Consequential, if the order selection map is chosen the same for both direction, the novel approach comes down exactly to the method of Maurer *et al.* [MSB17b].

5 Evaluation

5.1 Parameter Optimization

5.1.1 Overview

Due to the sophisticated concepts integrated in this approach, there are a lot of parameters that need to be chosen. There are 6 parameters that are individually optimized for each benchmark:

- α balances the data term and the regularization term and appears already in the method of Horn and Schunck [HS81] in Equation (2.8).
- γ as introduced in Equation (2.55) balances the brightness and gradient constancy assumption.
- β is used in coupling approaches to balance the similarity term S_2 against the regularization term of the auxiliary variables S_{aux}. It is first mentioned in Equation (2.93).
- λ is a parameter of the sigmoid function that is used to estimate the order selection map. It is introduced in Equation (3.8).
- T_1 and T_2 are thresholds making sure of a minimum average benefit [MSB17b] that the second-order approach needs to have over the first-order approach in order to be selected.

Additionally, numerous parameters are chosen fixed. Since the warping strategy makes use of an image pyramid, a downsampling factor is needed, which relates consecutive image resolutions. In this case, it was chosen to be $\eta = 0.95$. The number of iterations used was set to 5 outer iteration steps for the lagged non-linearity method and to 20 inner steps for the inner solver. For this, SOR was used with a over-relaxation parameter of 1.85. For the window size of the non-local selection approach, a 3×3 window was chosen. The integration constant that is needed for computing the regularization tensor, from which the directions $\mathbf{r}_1$ and $\mathbf{r}_2$ are retrieved, was set to $\rho = 1.5$.

As a sub-quadratic penalizer Ψ_D for the data term, the Charbonnier penalizer was used. The smoothness term utilizes two penalizers: For the dominant $\mathbf{r}_1$-direction, the Perona-Malik penalizer was used, while the second direction was penalized using Charbonnier again. In all penalizers, the small constant ϵ was set to 0.01.

5.1.2 Optimization Strategies

In order to optimize the parameters for each benchmark, the optimizer of Stoll *et al.* [SVMB17] was used. This open-source optimization framework provides several already implemented optimization strategies, some of which are cascadic sampling, the Downhill Simplex method [NM65] and the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [HO01].

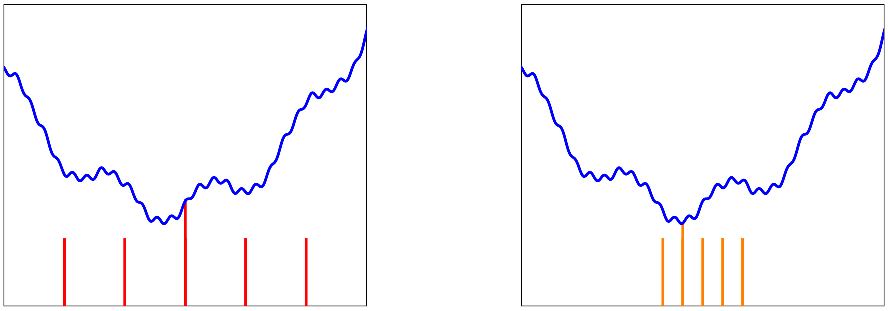


Figure 5.1: Schematic visualization of cascading sampling. A set of five equidistant samples and two cascades (left, right) is used. The individual optimum is extended to the function.

Cascadic Sampling The Cascadic sampling strategy has the goal of iterative refinement in order to find a global optimum. First, there is a fixed number of samples evaluated, which can either be equidistantly spaced in the search domain or evaluated on a logarithmic scale. After the optimum of these samples is found, the method cascades to a finer level. The new search space's minimum and maximum value is set to the sample's neighbors, such that the recent optimal value sits in the center of the new search space. Figure 5.1 visualizes the method schematically: In the first run (left), five sample positions are evaluated, with the optimum being the center (see extended mark). The neighbors of the optimum are then used as the new search space for the next cascade (right). In the end, a good approximation of the global minimum is found (extended orange mark).

The Downhill Simplex Method The Downhill Simplex method by Nelder and Mead [NM65] is based on a simplex in the parameter search space that evolves according to the local landscape and has the goal of contracting towards the global optimum. As desribed by Stoll *et al.* [SVMB17], the simplex is moved in a downhill manner by geometrically inspired steps called *reflection*, *expansion*, *contraction* and *reduction*.

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) Following Stoll *et al.* [SVMB17], the CMA-ES is a method with a stochastic heuristic. It was developed by Hansen and Ostermeier [HO01] and is based on an evolutionary algorithm that consists of the two iterated steps: *mutation* and *recombination*.

5.2 Benchmarking

5.2.1 Quality Measures

In order to assess the quality of an optical flow field, typically two measures are used. The *average endpoint error* (AEE) is defined as the average distance between the endpoint positions of the true flow vectors and the estimated flow vectors. For an image domain of size $M \times N$ and a ground truth flow $\mathbf{u}^{\text{gt}}$ and a flow estimate $\mathbf{u}^{\text{est}}$, it is computed as follows:

$$\text{AEE}(\mathbf{u}^{\text{gt}}, \mathbf{u}^{\text{est}}) = \frac{1}{MN} \cdot \sum_{i=0}^{M} \sum_{j=0}^{N} \left| \mathbf{u}_{i,j}^{\text{gt}} - \mathbf{u}_{i,j}^{\text{est}} \right| \ . \tag{5.1}$$

The *bad pixel error* (BP) is defined as the percentage of pixels, where the endpoint error is below a certain threshold T, usually 3 pixels. It is computed via

$$\mathrm{BP}(\mathbf{u}^{\mathrm{gt}}, \mathbf{u}^{\mathrm{est}}) = \frac{1}{MN} \cdot \sum_{i=0}^{M} \sum_{j=0}^{N} \left[\left| \mathbf{u}_{i,j}^{\mathrm{gt}} - \mathbf{u}_{i,j}^{\mathrm{est}} \right| > T \right] . \tag{5.2}$$

Both thresholds have in common that lower values correspond to a better quality.

5.2.2 Standard Benchmarks

In order to broadly compare optical flow estimation methods, a set of four benchmarks is commonly used for evaluation: the Middlebury [BSL+11], Sintel [BWSB12], KITTI12 [GLU12] and KITTI15 [MG15] dataset. Additionally, the classroom dataset as introduced by Maurer *et al.* [MSB17b] was used for evaluation.

One of the earlier datasets is the Middlebury benchmark by Baker *et al.* [BSL+11]. It contains 8 training sequences with a corresponding ground truth. However, a problem with the benchmark is that most of the image sequences have been created synthetically. Thus, they don't provide real-world challenges such as significant illumination changes, considerable out-of-plane motion or disturbances in data such as lens flares, under- and oversaturations or noise [SVMB17]. The dataset contains mostly fronto-parallel motion, thus it fits for first-order regularization approaches.

A more recent dataset is the MPI Sintel dataset [BWSB12]. It contains 23 sequences and over 1000 evaluation frames in total. The dataset was created from the animated open source short movie Sintel. Although it only contains computer-generated images, more challenges are provided, amongst them large non-rigid motions and more complexity due to blur, atmosphere and specular surfaces [BWSB12]. Although the dataset consists of different camera and object motions, the majority of the challenges can be considered as first-order motion.

A strong focus on automotive application is given in the KITTI datasets which were published in 2012 [GLU12] and 2015 [MG15]. The training set of the KITTI12 dataset consists of 194 grayscale training image sequences showing a mostly static scenery with ego motion. Since the images are real-world images, a ground truth could only be determined by a laser scanner, leading to a sparse ground truth for evaluation. While the KITTI12 variant has some shortcomings, its 2015 variant addresses some of them. It consists of 200 color image pairs showing highly dynamic scenes. Additionally, the individually moving objects could be recovered by fitting detailed CAD models to the point clouds in each frame [MG15], thus providing a dense ground truth for them. Due to the fact that the images were obtained using a camera mounted to a car, both datasets mostly contain affine motion challenges, which are related to second-order regularization.

The classroom dataset was introduced by Maurer *et al.* [MSB17b] and consists of four artificially created sequences. While the third sequence depicts an affine ego motion, the others contain fronto-parallel motion. Thus, the dataset is suitable to test a method that strives to perform well in first- and second-order scenarios.

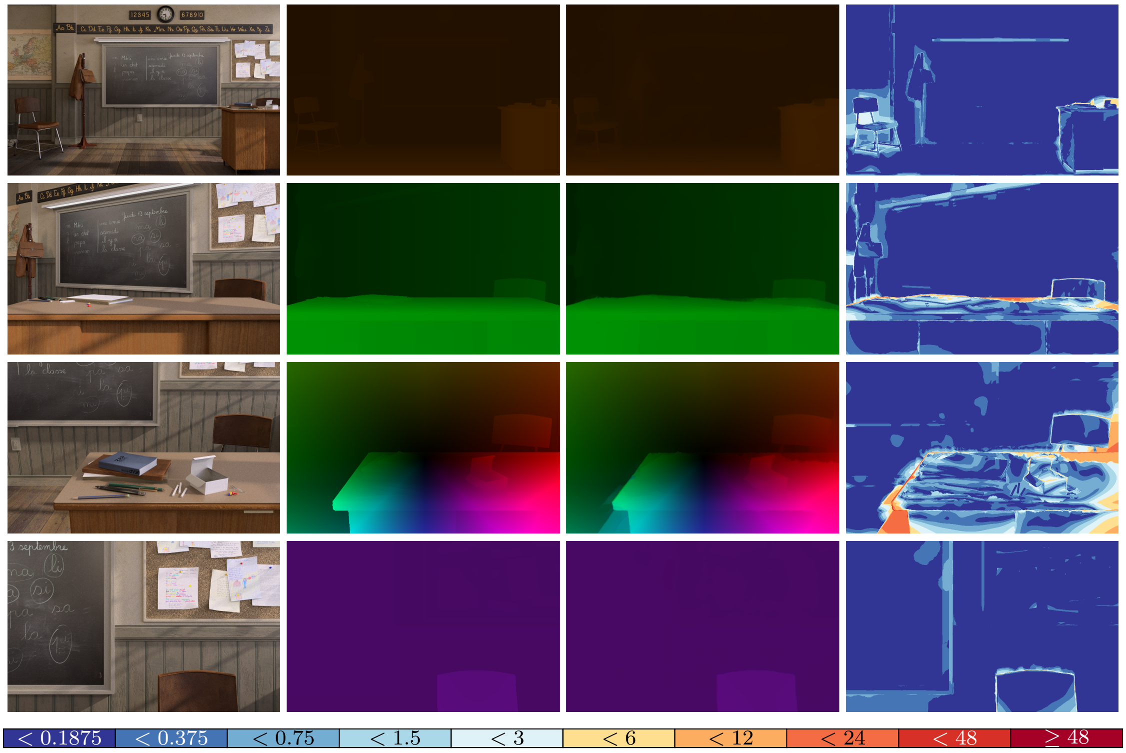


Figure 5.2: Results for the classroom sequence. *From top to bottom:* Sequences 1 to 4. *From left to right:* First frame, ground truth flow, estimated flow, bad pixel visualization. The bad pixel visualization shows the endpoint error per pixel using the color scale as shown below the images.

5.3 Results

In the following, the results of this thesis will be detailed. First, in order to directly assess the performance of the proposed method, an evaluation on the classroom dataset and on the commonly used benchmarks is given. Afterwards, a set of experiments working towards a more detailed understanding of the method is given. To this end, an analysis of the selected order is given, followed by an experiment on the quality of the order map and the experimental estimation of a theoretical maximum of the method. Finally, the parameter influence on the order map is examined.

5.3.1 Classroom Dataset

As a first evaluation, the performance of the novel method in first- and second-order scenarios was tested. Additionally, the results were compared to the base method by Maurer *et al.* [MSB17b]. Since it is well-suited as a test for order-adaptive approaches, the classroom dataset was used. In order to specifically compare the performance on the first- and second-order sequences, the result for each sequence individually is investigated, as well as the average on the whole dataset.

In order to find optimal values for the six free parameters as described in Section 5.1.1, a parameter optimization was done using all four sequences jointly.

	Classroom dataset				
	Seq. 1	Seq. 2	Seq. 3	Seq. 4	Avg.
isotropic	**0.116**	0.275	**0.737**	0.095	**0.307**
anisotropic	**0.116**	**0.254**	0.820	**0.089**	0.320

Table 5.1: Comparison of the isotropic order-adaptive approach of Maurer *et al.* [MSB17b] and the novel anisotropic variant for the classroom dataset in terms of the average endpoint error (AEE).

First, a visual examination of the method including the novel anisotropic order-selection scheme is given. Figure 5.2 visualizes the results of the proposed method showing the first image, the ground truth, the flow estimate and a bad pixel visualization. The flow visualizations are color-encoded using the method by Bruhn [Bru06] and the bad pixel visualization as first introduced in the KITTI15 dataset [MG15] plots the endpoint error per pixel using the logarithmic color scale as shown in Figure 5.2. While the method shows good results on Sequences 1, 2 and 4, there are several artifacts visible in Sequence 3. Since the third sequence is the only sequence containing affine motion, this experiment indicates that the novel approach performs good on sequences with fronto-parallel motion, but degrades on sequences with second-order motion.

The numerical results confirm this observation. Table 5.1 shows the endpoint error of the flow estimate and compares it to the isotropic method by Maurer *et al.* [MSB17b]. While a consistent result or improvement could be reached in Sequences 1, 2 and 4, Sequence 3 shows a degradation compared to the base method. This in turn results in a worse average endpoint error compared to the base method.

5.3.2 Benchmark Results

As a second evaluation, the presented method was evaluated on the most common benchmarks. Since all state-of-the-art methods are evaluated on them, this leads to a good comparison. The most important comparison is between the novel method and the base method by Maurer *et al.* [MSB17b], since a relative improvement to the base method would lead to a contribution to the state of the art, since the order-selection approach is commonly used in recent methods such as ProFlow [MB18].

For the numerical benchmark evaluation, the following settings were assigned: In the Middlebury dataset evaluation, Frame 10 and 11 of the 8 training sequences were used. The evaluation on the Sintel dataset was done using the clean pass and for each of the KITTI datasets, the image sequence of the first camera was used and evaluated using the *flow_occ* ground truth, which contains a ground truth even for occluded areas. Again, the six free parameters were optimized, which was done for each dataset individually. In the Sintel and both KITTI dataset evaluations, a subset of 8 frames was used for the parameter optimization, while evaluation was done using the whole set.

First, a visual evaluation for one example sequence from each benchmark is given in Figure 5.3. The Middlebury and Sintel examples show an overall good result. However, a drawback is visible, since in both examples a small gap is erroneously closed. On the KITTI benchmarks examples the method yields in some areas good results, but shows problems on the image boundaries and for the KITTI15 dataset in the shadow of the moving car.

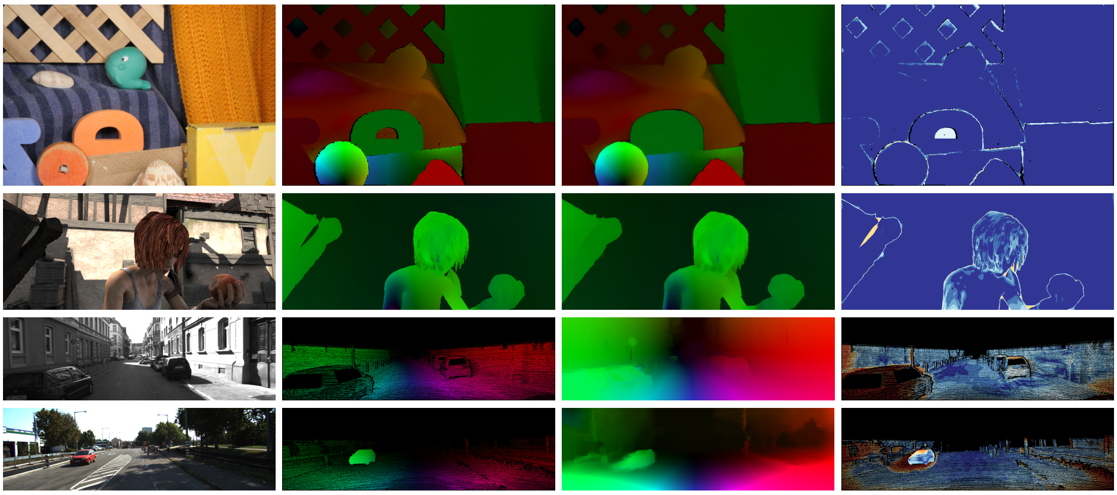


Figure 5.3: Visual results on the four benchmarks. *From left to right:* First image, ground truth flow field, estimated flow, bad pixel visualization. *From top to bottom:* Middlebury (RubberWhale Sequence, Frame 10), Sintel (alley_1 Sequence, Frame 1), KITTI12 (Sequence 15) and KITTI15 (Sequence 88) dataset.

	Middlebury	Sintel		KITTI 12		KITTI 15	
	AEE	AEE	BP	AEE	BP	AEE	BP
isotropic	**0.211**	**4.165**	9.231	**2.244**	**9.477**	**8.846**	**21.671**
anisotropic	**0.211**	4.205	**9.192**	2.597	10.180	9.138	22.183

Table 5.2: Comparison of the isotropic order-adaptive approach of Maurer *et al.* [MSB17b] and the newly presented anisotropic variant for the most common benchmarks.

The numerical results of the benchmark evaluation are given in Table 5.2, which shows a direct comparison of the anisotropic order selection method as presented in Chapter 4 and the base method by Maurer *et al.* [MSB17b]. The Middlebury dataset shows consistent results for both methods, while on the Sintel dataset, a small improvement with respect to the bad pixel error could be reached in conjunction with a small degradation with respect to the average endpoint error. At the same time, both KITTI datasets showed a distinct degradation for the novel method compared to its isotropic base.

The evaluation reveals that for the Middlebury and Sintel datasets, which contain mostly first-order scenarios, a small improvement or unchanged results could be shown. At the same time, the novel method seems to break in second-order scenarios which the KITTI datasets contain. This observation is consistent with the first evaluation, where similarly a small improvement in first-order sequences came along with a degradation in the second-order sequence. In summary, the proposed method only bears a small advantage over the method of Maurer *et al.* [MSB17b] in the case of first-order scenarios, but the goal of being robustly adaptive to all different kinds of scenarios could not be reached.

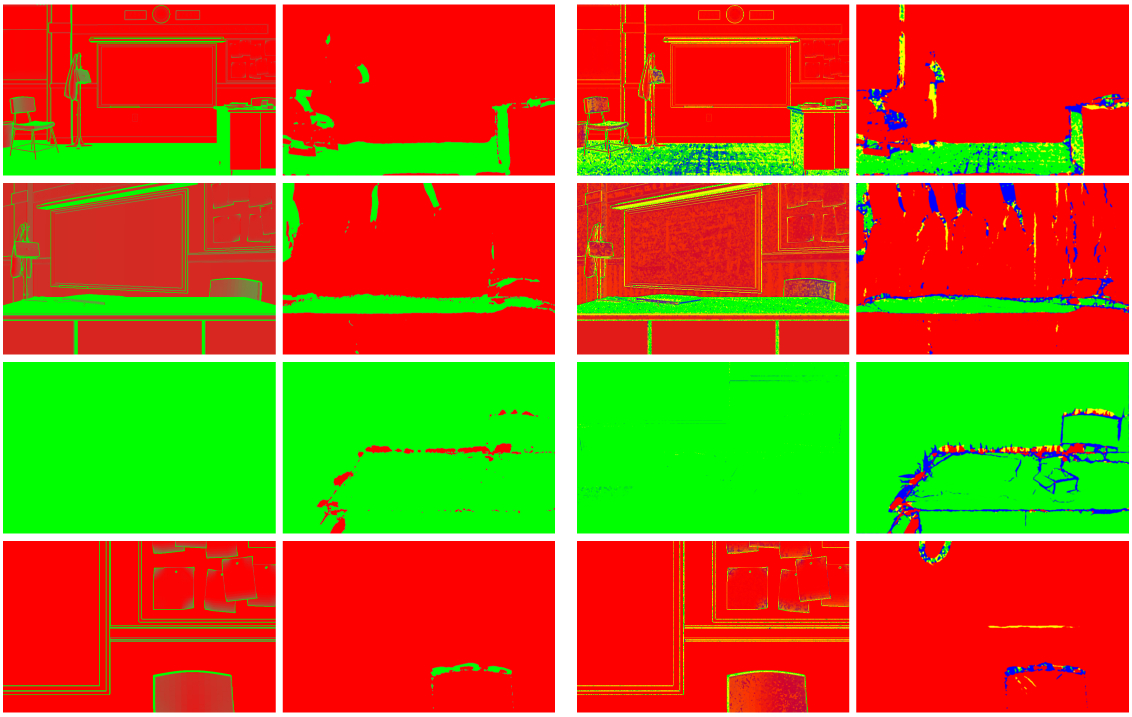


Figure 5.4: Comparison of color-coded order maps. Red: first order in both directions, green: second order in both directions, blue: first order in $\mathbf{r}_1$-direction, second order in $\mathbf{r}_2$-direction, yellow: second order in $\mathbf{r}_1$-direction, first order in $\mathbf{r}_2$-direction. *From left to right:* Gradient magnitude of ground truth flow, estimated order map by Maurer *et al.* [MSB17b], direction-based gradient magnitude, estimated anisotropic order map. *From top to bottom:* Sequence 1 to 4 of the classroom dataset.

5.3.3 Order Map Analysis

Since the estimated order maps are the core of the order selection method, they will be further investigated. Of special interest is the question, in which situations which regularization order is chosen and if the order selection maps approximate the order of the flow field as desired.

In the following, the resulting values of $\mathbf{c}(\mathbf{x})$ will be investigated. Since there is no direct pixelwise ground truth information on the order of a flow field, a comparison needs to be done using an approximation of the order. For this, as Maurer *et al.* [MSB17b] describe, the gradient magnitude of the ground truth flow field yields small values for fronto-parallel motion and large values for affine motion and thus may serve as a rough indicator which type of motion is present. The gradient magnitude is then compared to the resulting resulting order maps:

$$c \approx \sqrt{|\nabla u|^2 + |\nabla v|^2}\ . \tag{5.3}$$

Similarly, this comparison approach can be extended to the anisotropic order selection method by considering the magnitude of the directional gradient in $\mathbf{r}_1$- and $\mathbf{r}_2$-direction:

$$c_1 \approx \sqrt{(\mathbf{r}_1^\top \nabla u)^2 + (\mathbf{r}_1^\top \nabla v)^2}\ , \tag{5.4}$$

$$c_2 \approx \sqrt{(\mathbf{r}_2^\top \nabla u)^2 + (\mathbf{r}_2^\top \nabla v)^2}\ . \tag{5.5}$$

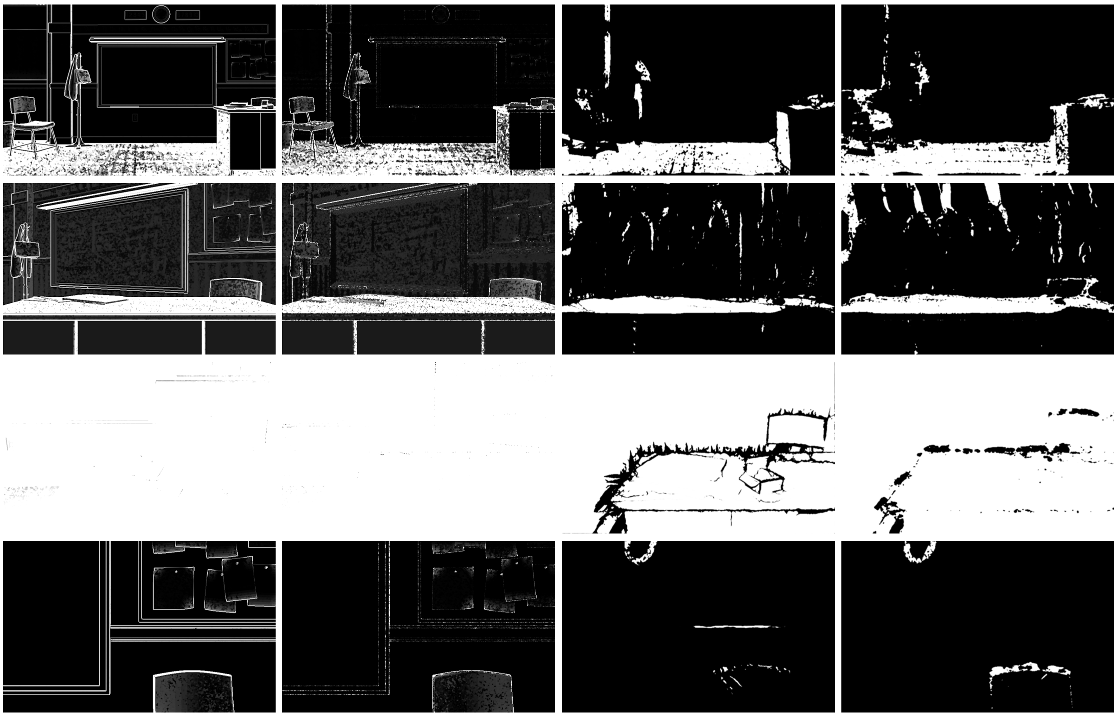


Figure 5.5: Component-wise visualization of order maps. *From left to right:* Ground truth approximation of c_1, ground truth approximation of c_2, calculated c_1, calculated c_2. *From top to bottom:* Sequence 1 to 4 of the classroom dataset. Black corresponds to first-order, white to second-order regularization.

Figure 5.4 gives an extensive visualization of the order maps of the classroom sequences which are shown row-wise. The first two columns describe the analysis by Maurer *et al.* [MSB17b]. While the first column shows the gradient magnitude where green is large and red is small, the second column depicts the estimated order map, with green as second order and red as first order. For the novel method, a similar comparison was done in the last two columns. The color-coding used here extends the red and green color, which now refers to first and second order in both directions. Additionally, blue refers to first order in the dominant direction, and second order in the other, while yellow is associated with the opposite. First, the expected order map is shown in the third column as approximated by Equations (5.4) and (5.5). For this, the computed vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ had to be used, which explains why the result is not perfectly clean but contains a small amount of noise. Then this map can be compared to the order map estimated by the method, which is shown in the last column. In order to have a clearer view on the individual components of the anisotropic order map, Figure 5.5 shows the estimated ground truth in the first two columns and the computed order map in the last columns component-wise.

It becomes obvious that the gradient magnitude for the base method and the directional gradient magnitude for the anisotropic method show strong correlations to the corresponding order maps. Thus, both methods roughly approximate the expected values. However, the results also shows differences and visible artifacts compared to the gradient magnitude and some observations can be noted.

	Classroom dataset				
	Seq. 1	Seq. 2	Seq. 3	Seq. 4	Avg.
isotropic	0.116	0.279	0.738	0.095	0.307
using ground truth	**0.103**	**0.270**	**0.675**	**0.086**	**0.284**
anisotropic	0.116	0.254	0.820	0.089	0.320
using ground truth	**0.104**	**0.234**	**0.688**	**0.086**	**0.278**

Table 5.3: Results of the order map quality test. Comparison of the AEE of the isotropic and anisotropic version of the method and its modification that the gradient magnitude of the ground truth flow field.

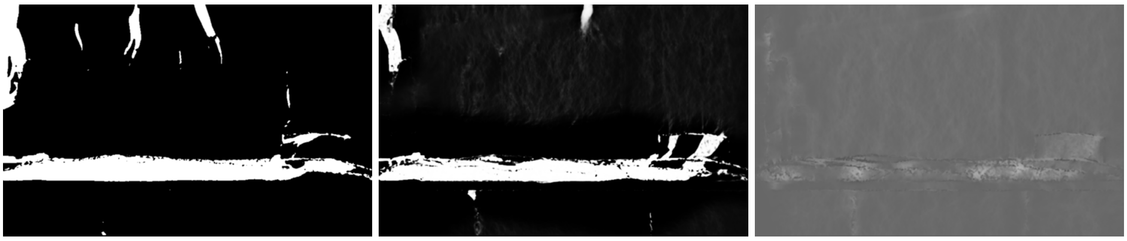
A first observation lies in the newly introduced possibility of choosing a different regularization order in the two directions. From the gradient magnitude estimation in the third column of Figure 5.4 it can be seen that selecting a different order is not required in many situations. The differently selected order, which is visualized by blue and yellow only comes down to very few pixel locations. However, as a view on the actually calculated order map in the fourth column shows, the proposed method shows many cases where two different orders are selected, too many compared to the gradient magnitude map.

A second observation can be made when comparing the anisotropic order selection method to the isotropic method. Since the isotropic order map is related to the two anisotropic maps, the three maps in the second column of Figure 5.4 and the third and forth column of Figure 5.5 can be compared. Visually inspecting them yields a strong similarity of the isotropic c with the second component of the anisotropic case, c_2. Thus, it seems like the first component in the dominant direction $\mathbf{r}_1$ has only very little influence on the isotropic order selection map. Additionally, especially in the third sequence which contains second-order motion, the order map c_1 introduces spiky artifacts. This could be a reason for observation that the introduction of anisotropy reduces the performance in second-order scenarios as described in Sections 5.3.1 and 5.3.2.

5.3.4 Order Map Quality Test

The previous section made use of the gradient magnitude as an expected outcome of the order selection maps. In order to assess, if working towards this desired behavior results in a performance improvement, an additional test is executed. In the following, a goal is to also estimate the theoretical result that can be achieved, when a high-quality order map is used, where a high quality means being close to the gradient magnitude. For this experiment, the estimated gradient magnitude map, as depicted in Figures 5.4 and 5.5, is used inside the optical flow estimation method instead of computing the order map. The expected outcome of this test is an improvement of the resulting error measures up to a theoretical maximum which could be achieved through the order maps. Since this evaluation is novel for the isotropic as well as the anisotropic case, it is realized for both using the classroom dataset.

In order to perform this experiment, the ground truth flow field is read in during the optical flow estimation and downsampled to the corresponding warping layer. Then using Equation (5.3) for the isotropic case and Equations (5.4) and (5.5) for the anisotropic case, a gradient magnitude map is



(a) $\lambda = 1.1 \cdot 10^{-6}$ **(b)** $\lambda = 8.02 \cdot 10^{-5}$ **(c)** $\lambda = 6.53 \cdot 10^{-4}$

Figure 5.6: Order selection map c_1 for different choices of λ. Black corresponds to first-order, white to second-order selection.

computed, which will be regarded as a near ground truth estimate. In order to be similar to the calculated order maps, the gradient magnitude map is scaled by a constant factor of 20 and clipped to the range $[0; 1]$. In the following calculation steps, this map is then used instead of the original order maps. For the evaluation, the parameters were again optimized on the dataset, individually for the isotropic and anisotropic case.

Table 5.3 shows the test results. As can immediately be seen, the inclusion of ground truth flow information improved the error for all sequences individually and for the whole dataset, both for the isotropic and the anisotropic case. Additionally, an advantage of the anisotropic selection scheme over the isotropic case can be observed. Although such a clear advantage is only observed for this theoretical evaluation, it still gives a hint that the anisotropic variant might be superior to the simpler isotropic method. Also investigating ways to keep the estimated order map closer to its theoretical ground truth seems to be a good way to move forward.

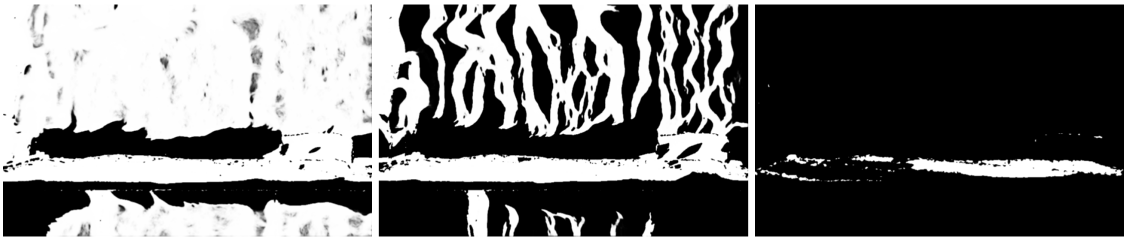
5.3.5 Parameter Influence on the Order Selection

Since the estimated order maps are central to the here proposed method, the parameters influencing them will be analyzed in more detail. The parameters having the most direct influence on the order maps are λ and T_1 respective T_2.

Influence of λ First, the influence of the parameter λ will be determined. In the energy functional as shown in Equation (4.16), it controls the influence of the selection term $\phi(c)$. In order to visualize this influence, the resulting order selection map for several choices of λ is calculated for Sequence 2 of the classroom dataset.

Figure 5.6 shows the resulting map for three different choices of λ. As already shown in Figure 3.1 and also depicted here, the parameter λ controls the contrast of the order selection map. For a large λ, the contrast is small, and the resulting value is very close to 0.5. A large choice on the other hand gives a hard separation between first and second order.

This can also be understood from a second point of view: As described before, in the regularization term as shown in Equation (3.22), the function $\phi(c)$ is weighted with the parameter λ. The definition of $\phi(c)$ is $\phi(c) = c \cdot \ln(c) + (1 - c) \cdot \ln(1 - c)$, with the corresponding plot shown in Figure 3.2, which has its global minimum at 0.5. This again explains that the choice of a high λ implies that



(a) $T_1 = 1.0 \cdot 10^{-7}$ **(b)** $T_1 = 2.92 \cdot 10^{-5}$ **(c)** $T_1 = 4.31 \cdot 10^{-2}$

Figure 5.7: Order selection map c_1 for different choices of T_1. Black corresponds to first-order, white to second-order selection.

the order map is close to the minimum of 0.5 in order to reach the minimum of the functional. Additionally, the function ϕ can be regarded as a prior on the solution whose strength is controlled by λ.

Influence of T_1 and T_2 Second, the parameters that have a strong influence on the order selection map are the parameters T_1 and T_2, which are selected individually per map. As described before, for each map, the parameter is a threshold that needs to be exceeded by the energy in order to select second order over first order. This has the background that first-order regularization can be regarded as a subset of second-order regularization; enforcing a linearity also includes the possibility of enforcing constancy. Similar to before, the resulting order selection map of Sequence 3 from the classroom dataset is calculated for several choices of T_1.

Figure 5.7 shows three resulting order selection maps c_1. As expected, choosing small values (i.e. omitting the threshold) results in very often selecting the second-order method. On the other hand, a large value reduces the areas of second order. For even larger values than depicted, the resulting order map would be completely black, i.e. globally first-order regularization would be chosen.

6 Conclusion

This thesis addresses the problem of recent variational methods with order-adaptive regularization that are restricted to an isotropic order selection schemes. So far, order-adaptive regularization approaches as introduced by Maurer *et al.* [MSB17b] select a single regularization order per pixel. Although the underlying regularization schemes are anisotropic and thus adaptive to image directions, the corresponding order selection scheme in recent approaches is restricted to an isotropic formulation.

This thesis strives to overcome this shortcoming. First, all steps of the isotropic order selection scheme by Maurer *et al.* [MSB17b] were derived. Since not only the model and minimzation was given, but also the discretization scheme, the structure of the fully discretized linearized system could be analyzed and the block structure of the resulting sparse matrix could be shown. Afterwards, the model by Maurer *et al.* [MSB17b] was extended to selecting the order adaptively in each local image direction. By determining the minimization of the novel method, it was shown that the theory of Maurer *et al.* [MSB17b] still holds for this extension and the implementation is straightforward. Finally in order to assess its quality, the evaluation and analysis of the newly presented method was given. The evaluation on the classroom dataset and on the commonly used benchmarks showed that the anisotropic order selection approach gives a small improvement on fronto-parallel scenarios. However, for sequences containing second-order affine motion such as the KITTI sequence, this method does not seem applicable.

To gain more insight, an analysis of the resulting order selection maps was included. In comparison with the resulting order map of the isotropic approach it was shown that the selected order in the secondary direction has a strong similarity to the isotropic order map. At the same time, the order map for the primary, dominant direction introduced artifacts, especially in second-order scenarios. Finally it was shown that when using the oriented gradient magnitude of the ground truth flow field, the result is a strong improvement of the corresponding error measures. This experimental evaluation gives a hint towards the theoretically possible improvement that can be reached through the order selection map. Additionally, it was shown that the anisotropic variant clearly outperformed the isotropic order selection approach in this theoretical setting.

This thesis showed that introducing anisotropy into the order selection process using the presented method brings a small advantage on some error measures of benchmarks with first-order motion scenarios. On benchmarks depicting second-order motion scenarios such as the KITTI datasets the performance degrades. Additionally it was proven that an improved order map quality would yield better results and that for an improved order map quality there might be an advantage of the anisotropic order selection of the isotropic variant.

6.1 Outlook

Although this first implementation of anisotropic selection schemes did not lead to an improvement on all benchmarks, the results showed that a further development of the method could still lead to a measurable advancement.

Quality Improvement of the Order Maps As shown for the classroom sequence in Section 5.3.4, the usage of the gradient magnitude led to a solid improvement of the resulting flow field for the isotropic as well as the anisotropic case. Thus, investigating ways to estimate the resulting order map closer to the gradient magnitude could prove beneficial. Starting points for this could be revisiting the integration size of the non-local approach, a different formulation of calculating the order map or a different integration of the current flow estimate into the estimation of the order maps.

Regularization of the Order Maps As described in Section 5.3.3, many cases could be observed where a different regularization order was chosen in the two directions, although this was not required. To avoid this behavior, an additional regularization that enforces a stronger similarity of the two maps could be integrated into the method and further push the benchmark results. This would however increase the complexity since one of the nice benefits is that the calculation of the order map can currently be done explicitly. Introducing additional regularization could change this.

Regularization Vectors The directions that are retrieved as the eigenvectors from the regularization tensor are used several times in the novel approach. Compared to the anisotropic regularization term, where they are used as a directional derivative, the anisotropic order selection method uses them directly. Thus, the quality of the directional information has a direct influence on the quality of the method. However, the normalized vectors have the drawback of being prone to noise in flat image areas, where their eigenvalue would be small. So, research in the direction of a different integration of them into the order selection method, e.g. by additionally incorporating their corresponding eigenvalue may be helpful. Also, investigating different integration sizes ρ to compute the regularization tensor could bear potential.

A Stencils

A.1 Anisotropic Diffusion Stencil

As described in detail in Section 3.3.1, a discrete stencil derivation for an anisotropic diffusion process is executed. Given the diffusion expression $\operatorname{div}(\mathbf{D} \cdot \nabla u) = 0$ with the spatially dependent diffusion tensor $\mathbf{D}$ with

$$\mathbf{D} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} , \tag{A.1}$$

then the entries of the spatially dependent 3×3 stencil $\mathcal{S}^1$ for the position (i, j) are given below.

Note that in the stencil computation, the diffusion tensor entries a, b, c and the parameters α and β are needed at positions $\left(i + \frac{1}{2}, j + \frac{1}{2}\right)$. This can be achieved by a simple average of the surrounding values. However, for readability purposes in the following the variables are referenced by their integer position as an abbreviation for the subpixel location, e.g. $\alpha_{i,j}$ abbreviates $\alpha_{i+\frac{1}{2},j+\frac{1}{2}}$.

Stencil weight for $u_{i+1,j+1}$:

$$\begin{aligned}
&- \frac{\alpha_{i,j} \cdot \chi_{i,j+1} \cdot \chi_{i+1,j+1} \cdot c_{i,j}}{h_y^2} - \frac{\alpha_{i,j} \cdot \chi_{i+1,j} \cdot \chi_{i+1,j+1} \cdot a_{i,j}}{h_x^2} - \frac{0.5 \cdot \beta_{i,j} \cdot \chi_{i,j+1} \cdot \chi_{i+1,j+1} \cdot b_{i,j}}{h_x h_y} \\
&- \frac{0.5 \cdot \beta_{i,j} \cdot \chi_{i+1,j} \cdot \chi_{i+1,j+1} \cdot b_{i,j}}{h_x h_y} - \frac{0.5 \cdot \chi_{i,j+1} \cdot \chi_{i+1,j+1} \cdot b_{i,j}}{h_x h_y} - \frac{0.5 \cdot \chi_{i+1,j} \cdot \chi_{i+1,j+1} \cdot b_{i,j}}{h_x h_y}
\end{aligned}$$

Stencil weight for $u_{i+1,j}$:

$$\begin{aligned}
&\frac{\alpha_{i,j-1} \cdot \chi_{i,j} \cdot \chi_{i+1,j} \cdot c_{i,j-1}}{h_y^2} + \frac{\alpha_{i,j-1} \cdot \chi_{i+1,j}^2 \cdot a_{i,j-1}}{h_x^2} + \frac{\alpha_{i,j} \cdot \chi_{i,j+1} \cdot \chi_{i+1,j+1} \cdot c_{i,j}}{h_y^2} \\
&+ \frac{\alpha_{i,j} \cdot \chi_{i+1,j}^2 \cdot a_{i,j}}{h_x^2} + \frac{0.5 \cdot \beta_{i,j-1} \cdot \chi_{i,j} \cdot \chi_{i+1,j} \cdot b_{i,j-1}}{h_x h_y} + \frac{0.5 \cdot \beta_{i,j-1} \cdot \chi_{i+1,j}^2 \cdot b_{i,j-1}}{h_x h_y} \\
&+ \frac{0.5 \cdot \beta_{i,j} \cdot \chi_{i,j+1} \cdot \chi_{i+1,j} \cdot b_{i,j}}{h_x h_y} + \frac{0.5 \cdot \beta_{i,j} \cdot \chi_{i+1,j} \cdot \chi_{i+1,j+1} \cdot b_{i,j}}{h_x h_y} + \frac{0.5 \cdot \chi_{i,j} \cdot \chi_{i+1,j} \cdot b_{i,j-1}}{h_x h_y} \\
&- \frac{0.5 \cdot \chi_{i,j+1} \cdot \chi_{i+1,j} \cdot b_{i,j}}{h_x h_y} - \frac{\chi_{i+1,j}^2 \cdot a_{i,j-1}}{h_x^2} - \frac{\chi_{i+1,j}^2 \cdot a_{i,j}}{h_x^2} \\
&- \frac{0.5 \cdot \chi_{i+1,j}^2 \cdot b_{i,j-1}}{h_x h_y} + \frac{0.5 \cdot \chi_{i+1,j} \cdot \chi_{i+1,j+1} \cdot b_{i,j}}{h_x h_y}
\end{aligned}$$

Stencil weight for $u_{i+1,j-1}$:

$$\begin{aligned}
&-\frac{\alpha_{i,j-1}\cdot\chi_{i,j}\cdot\chi_{i+1,j}\cdot c_{i,j-1}}{h_y^2}-\frac{\alpha_{i,j-1}\cdot\chi_{i+1,j-1}\cdot\chi_{i+1,j}\cdot a_{i,j-1}}{h_x^2}\\
&-\frac{0.5\cdot\beta_{i,j-1}\cdot\chi_{i,j}\cdot\chi_{i+1,j-1}\cdot b_{i,j-1}}{h_xh_y}-\frac{0.5\cdot\beta_{i,j-1}\cdot\chi_{i+1,j}^2\cdot b_{i,j-1}}{h_xh_y}+\frac{0.5\cdot\chi_{i,j}\cdot\chi_{i+1,j-1}\cdot b_{i,j-1}}{h_xh_y}\\
&+\frac{0.5\cdot\chi_{i+1,j}^2\cdot b_{i,j-1}}{h_xh_y}
\end{aligned}$$

Stencil weight for $u_{i,j+1}$:

$$\begin{aligned}
&\frac{\alpha_{i-1,j}\cdot\chi_{i,j}\cdot\chi_{i,j+1}\cdot a_{i-1,j}}{h_x^2}+\frac{\alpha_{i-1,j}\cdot\chi_{i,j+1}^2\cdot c_{i-1,j}}{h_y^2}+\frac{\alpha_{i,j}\cdot\chi_{i,j+1}^2\cdot c_{i,j}}{h_y^2}\\
&+\frac{\alpha_{i,j}\cdot\chi_{i+1,j}\cdot\chi_{i+1,j+1}\cdot a_{i,j}}{h_x^2}+\frac{0.5\cdot\beta_{i-1,j}\cdot\chi_{i,j}\cdot\chi_{i,j+1}\cdot b_{i-1,j}}{h_xh_y}+\frac{0.5\cdot\beta_{i-1,j}\cdot\chi_{i,j+1}^2\cdot b_{i-1,j}}{h_xh_y}\\
&+\frac{0.5\cdot\beta_{i,j}\cdot\chi_{i,j+1}\cdot\chi_{i+1,j}\cdot b_{i,j}}{h_xh_y}+\frac{0.5\cdot\beta_{i,j}\cdot\chi_{i,j+1}\cdot\chi_{i+1,j+1}\cdot b_{i,j}}{h_xh_y}+\frac{0.5\cdot\chi_{i,j}\cdot\chi_{i,j+1}\cdot b_{i-1,j}}{h_xh_y}\\
&-\frac{0.5\cdot\chi_{i,j+1}^2\cdot b_{i-1,j}}{h_xh_y}-\frac{\chi_{i,j+1}^2\cdot c_{i-1,j}}{h_y^2}-\frac{\chi_{i,j+1}^2\cdot c_{i,j}}{h_y^2}\\
&-\frac{0.5\cdot\chi_{i,j+1}\cdot\chi_{i+1,j}\cdot b_{i,j}}{h_xh_y}+\frac{0.5\cdot\chi_{i,j+1}\cdot\chi_{i+1,j+1}\cdot b_{i,j}}{h_xh_y}
\end{aligned}$$

Stencil weight for $u_{i,j}$:

$$\begin{aligned}
&-\frac{\alpha_{i-1,j-1}\cdot\chi_{i,j}^2\cdot a_{i-1,j-1}}{h_x^2}-\frac{\alpha_{i-1,j-1}\cdot\chi_{i,j}^2\cdot c_{i-1,j-1}}{h_y^2}-\frac{\alpha_{i-1,j}\cdot\chi_{i,j}^2\cdot a_{i-1,j}}{h_x^2}\\
&-\frac{\alpha_{i-1,j}\cdot\chi_{i,j+1}^2\cdot c_{i-1,j}}{h_y^2}-\frac{\alpha_{i,j-1}\cdot\chi_{i,j}^2\cdot c_{i,j-1}}{h_y^2}-\frac{\alpha_{i,j-1}\cdot\chi_{i+1,j}^2\cdot a_{i,j-1}}{h_x^2}\\
&-\frac{\alpha_{i,j}\cdot\chi_{i,j+1}^2\cdot c_{i,j}}{h_y^2}-\frac{\alpha_{i,j}\cdot\chi_{i+1,j}^2\cdot a_{i,j}}{h_x^2}-\frac{\beta_{i-1,j-1}\cdot\chi_{i,j}^2\cdot b_{i-1,j-1}}{h_xh_y}\\
&-\frac{\beta_{i-1,j}\cdot\chi_{i,j}\cdot\chi_{i,j+1}\cdot b_{i-1,j}}{h_xh_y}-\frac{\beta_{i,j-1}\cdot\chi_{i,j}\cdot\chi_{i+1,j}\cdot b_{i,j-1}}{h_xh_y}-\frac{\beta_{i,j}\cdot\chi_{i,j+1}\cdot\chi_{i+1,j}\cdot b_{i,j}}{h_xh_y}\\
&+\frac{\chi_{i,j}^2\cdot a_{i-1,j-1}}{h_x^2}+\frac{\chi_{i,j}^2\cdot a_{i-1,j}}{h_x^2}+\frac{\chi_{i,j}^2\cdot b_{i-1,j-1}}{h_xh_y}\\
&+\frac{\chi_{i,j}^2\cdot c_{i-1,j-1}}{h_y^2}+\frac{\chi_{i,j}^2\cdot c_{i,j-1}}{h_y^2}-\frac{\chi_{i,j}\cdot\chi_{i,j+1}\cdot b_{i-1,j}}{h_xh_y}\\
&-\frac{\chi_{i,j}\cdot\chi_{i+1,j}\cdot b_{i,j-1}}{h_xh_y}+\frac{\chi_{i,j+1}^2\cdot c_{i-1,j}}{h_y^2}+\frac{\chi_{i,j+1}^2\cdot c_{i,j}}{h_y^2}\\
&+\frac{\chi_{i,j+1}\cdot\chi_{i+1,j}\cdot b_{i,j}}{h_xh_y}+\frac{\chi_{i+1,j}^2\cdot a_{i,j-1}}{h_x^2}+\frac{\chi_{i+1,j}^2\cdot a_{i,j}}{h_x^2}
\end{aligned}$$

Stencil weight for $u_{i,j-1}$:

$$\begin{aligned}
&\frac{\alpha_{i-1,j-1} \cdot \chi_{i,j-1} \cdot \chi_{i,j} \cdot a_{i-1,j-1}}{h_x^2} + \frac{\alpha_{i-1,j-1} \cdot \chi_{i,j}^2 \cdot c_{i-1,j-1}}{h_y^2} + \frac{\alpha_{i,j-1} \cdot \chi_{i,j}^2 \cdot c_{i,j-1}}{h_y^2} \\
&+ \frac{\alpha_{i,j-1} \cdot \chi_{i+1,j-1} \cdot \chi_{i+1,j} \cdot a_{i,j-1}}{h_x^2} + \frac{0.5 \cdot \beta_{i-1,j-1} \cdot \chi_{i,j-1} \cdot \chi_{i,j} \cdot b_{i-1,j-1}}{h_x h_y} \\
&+ \frac{0.5 \cdot \beta_{i-1,j-1} \cdot \chi_{i,j}^2 \cdot b_{i-1,j-1}}{h_x h_y} + \frac{0.5 \cdot \beta_{i,j-1} \cdot \chi_{i,j} \cdot \chi_{i+1,j-1} \cdot b_{i,j-1}}{h_x h_y} \\
&+ \frac{0.5 \cdot \beta_{i,j-1} \cdot \chi_{i,j} \cdot \chi_{i+1,j} \cdot b_{i,j-1}}{h_x h_y} + \frac{0.5 \cdot \chi_{i,j-1} \cdot \chi_{i,j} \cdot b_{i-1,j-1}}{h_x h_y} - \frac{0.5 \cdot \chi_{i,j}^2 \cdot b_{i-1,j-1}}{h_x h_y} \\
&- \frac{\chi_{i,j}^2 \cdot c_{i-1,j-1}}{h_y^2} - \frac{\chi_{i,j}^2 \cdot c_{i,j-1}}{h_y^2} - \frac{0.5 \cdot \chi_{i,j} \cdot \chi_{i+1,j-1} \cdot b_{i,j-1}}{h_x h_y} \\
&+ \frac{0.5 \cdot \chi_{i,j} \cdot \chi_{i+1,j} \cdot b_{i,j-1}}{h_x h_y}
\end{aligned}$$

Stencil weight for $u_{i-1,j+1}$:

$$\begin{aligned}
&-\frac{\alpha_{i-1,j} \cdot \chi_{i-1,j+1} \cdot \chi_{i,j+1} \cdot c_{i-1,j}}{h_y^2} - \frac{\alpha_{i-1,j} \cdot \chi_{i,j} \cdot \chi_{i,j+1} \cdot a_{i-1,j}}{h_x^2} \\
&- \frac{0.5 \cdot \beta_{i-1,j} \cdot \chi_{i-1,j+1} \cdot \chi_{i,j} \cdot b_{i-1,j}}{h_x h_y} - \frac{0.5 \cdot \beta_{i-1,j} \cdot \chi_{i,j+1}^2 \cdot b_{i-1,j}}{h_x h_y} + \frac{0.5 \cdot \chi_{i-1,j+1} \cdot \chi_{i,j} \cdot b_{i-1,j}}{h_x h_y} \\
&+ \frac{0.5 \cdot \chi_{i,j+1}^2 \cdot b_{i-1,j}}{h_x h_y}
\end{aligned}$$

Stencil weight for $u_{i-1,j}$:

$$\begin{aligned}
&\frac{\alpha_{i-1,j-1} \cdot \chi_{i-1,j} \cdot \chi_{i,j} \cdot c_{i-1,j-1}}{h_y^2} + \frac{\alpha_{i-1,j-1} \cdot \chi_{i,j}^2 \cdot a_{i-1,j-1}}{h_x^2} + \frac{\alpha_{i-1,j} \cdot \chi_{i-1,j+1} \cdot \chi_{i,j+1} \cdot c_{i-1,j}}{h_y^2} \\
&+ \frac{\alpha_{i-1,j} \cdot \chi_{i,j}^2 \cdot a_{i-1,j}}{h_x^2} + \frac{0.5 \cdot \beta_{i-1,j-1} \cdot \chi_{i-1,j} \cdot \chi_{i,j} \cdot b_{i-1,j-1}}{h_x h_y} + \frac{0.5 \cdot \beta_{i-1,j-1} \cdot \chi_{i,j}^2 \cdot b_{i-1,j-1}}{h_x h_y} \\
&+ \frac{0.5 \cdot \beta_{i-1,j} \cdot \chi_{i-1,j+1} \cdot \chi_{i,j} \cdot b_{i-1,j}}{h_x h_y} + \frac{0.5 \cdot \beta_{i-1,j} \cdot \chi_{i,j} \cdot \chi_{i,j+1} \cdot b_{i-1,j}}{h_x h_y} \\
&+ \frac{0.5 \cdot \chi_{i-1,j} \cdot \chi_{i,j} \cdot b_{i-1,j-1}}{h_x h_y} - \frac{0.5 \cdot \chi_{i-1,j+1} \cdot \chi_{i,j} \cdot b_{i-1,j}}{h_x h_y} - \frac{\chi_{i,j}^2 \cdot a_{i-1,j-1}}{h_x^2} \\
&- \frac{\chi_{i,j}^2 \cdot a_{i-1,j}}{h_x^2} - \frac{0.5 \cdot \chi_{i,j}^2 \cdot b_{i-1,j-1}}{h_x h_y} + \frac{0.5 \cdot \chi_{i,j} \cdot \chi_{i,j+1} \cdot b_{i-1,j}}{h_x h_y}
\end{aligned}$$

Stencil weight for $u_{i-1,j-1}$:

$$-\frac{\alpha_{i-1,j-1} \cdot \chi_{i-1,j} \cdot \chi_{i,j} \cdot c_{i-1,j-1}}{h_y^2} - \frac{\alpha_{i-1,j-1} \cdot \chi_{i,j-1} \cdot \chi_{i,j} \cdot a_{i-1,j-1}}{h_x^2}$$
$$- \frac{0.5 \cdot \beta_{i-1,j-1} \cdot \chi_{i-1,j} \cdot \chi_{i,j} \cdot b_{i-1,j-1}}{h_x h_y} - \frac{0.5 \cdot \beta_{i-1,j-1} \cdot \chi_{i,j-1} \cdot \chi_{i,j} \cdot b_{i-1,j-1}}{h_x h_y}$$
$$- \frac{0.5 \cdot \chi_{i-1,j} \cdot \chi_{i,j} \cdot b_{i-1,j-1}}{h_x h_y} - \frac{0.5 \cdot \chi_{i,j-1} \cdot \chi_{i,j} \cdot b_{i-1,j-1}}{h_x h_y}$$

A.2 Anisotropic Coupled Diffusion Stencil

Given the coupled diffusion expression as presented in Equations (3.130) and (3.131), the corresponding nine stencils of $\mathcal{S}^2$ with a size of up to 3×3 are given below. The structure of the stencils is given in Figure 3.4.

As before, note that in the stencil computation, the diffusion tensor entries a, b, c and the parameters α and β are needed at in-between positions, but are abbreviated using their integer position.

A.2.1 Stencil $\mathcal{S}^{2,u,u}$

Due to the similar formulation of the two discretized diffusion expressions, this stencil is equivalent to the stencil $\mathcal{S}^1$ as shown in Appendix A.1.

A.2.2 Stencil $\mathcal{S}^{2,u,p}$

Stencil weight for $p_{i,j}$:

$$\frac{\chi_{i,j+1} \cdot b_{i,j}}{h_y} + \frac{\chi_{i+1,j} \cdot a_{i,j}}{h_x}$$

Stencil weight for $p_{i,j-1}$:

$$-\frac{\chi_{i,j} \cdot b_{i,j-1}}{h_y} + \frac{\chi_{i+1,j} \cdot a_{i,j-1}}{h_x}$$

Stencil weight for $p_{i-1,j}$:

$$-\frac{\chi_{i,j} \cdot a_{i-1,j}}{h_x} + \frac{\chi_{i,j+1} \cdot b_{i-1,j}}{h_y}$$

Stencil weight for $p_{i-1,j-1}$:

$$-\frac{\chi_{i,j} \cdot a_{i-1,j-1}}{h_x} - \frac{\chi_{i,j} \cdot b_{i-1,j-1}}{h_y}$$

A.2.3 Stencil $\mathcal{S}^{2,u,q}$

Stencil weight for $q_{i,j}$:

$$\frac{\chi_{i,j+1} \cdot c_{i,j}}{h_y} + \frac{\chi_{i+1,j} \cdot b_{i,j}}{h_x}$$

Stencil weight for $q_{i,j-1}$:

$$-\frac{\chi_{i,j} \cdot c_{i,j-1}}{h_y} + \frac{\chi_{i+1,j} \cdot b_{i,j-1}}{h_x}$$

Stencil weight for $q_{i-1,j}$:

$$-\frac{\chi_{i,j} \cdot b_{i-1,j}}{h_x} + \frac{\chi_{i,j+1} \cdot c_{i-1,j}}{h_y}$$

Stencil weight for $q_{i-1,j-1}$:

$$-\frac{\chi_{i,j} \cdot b_{i-1,j-1}}{h_x} - \frac{\chi_{i,j} \cdot c_{i-1,j-1}}{h_y}$$

A.2.4 Stencil $\mathcal{S}^{2,p,u}$

Stencil weight for $u_{i+1,j+1}$:

$$-\frac{\chi_{i+1,j+1} \cdot a_{i,j}}{h_x} - \frac{\chi_{i+1,j+1} \cdot b_{i,j}}{h_y}$$

Stencil weight for $u_{i+1,j}$:

$$-\frac{\chi_{i+1,j} \cdot a_{i,j}}{h_x} + \frac{\chi_{i+1,j+1} \cdot b_{i,j}}{h_y}$$

Stencil weight for $u_{i,j+1}$:

$$-\frac{\chi_{i,j+1} \cdot b_{i,j}}{h_y} + \frac{\chi_{i+1,j+1} \cdot a_{i,j}}{h_x}$$

Stencil weight for $u_{i,j}$:
$\frac{\chi_{i,j+1} \cdot b_{i,j}}{h_y} + \frac{\chi_{i+1,j} \cdot a_{i,j}}{h_x}$

A.2.5 Stencil $\mathcal{S}^{2,p,p}$

Stencil weight for $p_{i,j}$:
$2 \cdot a_{i,j}$

A.2.6 Stencil $\mathcal{S}^{2,p,q}$

Stencil weight for $q_{i,j}$:
$2 \cdot b_{i,j}$

A.2.7 Stencil $\mathcal{S}^{2,q,u}$

Stencil weight for $u_{i+1,j+1}$:
$-\frac{\chi_{i+1,j+1} \cdot b_{i,j}}{h_x} - \frac{\chi_{i+1,j+1} \cdot c_{i,j}}{h_y}$

Stencil weight for $u_{i+1,j}$:
$-\frac{\chi_{i+1,j} \cdot b_{i,j}}{h_x} + \frac{\chi_{i+1,j+1} \cdot c_{i,j}}{h_y}$

Stencil weight for $u_{i,j+1}$:
$-\frac{\chi_{i,j+1} \cdot c_{i,j}}{h_y} + \frac{\chi_{i+1,j+1} \cdot b_{i,j}}{h_x}$

Stencil weight for $u_{i,j}$:
$\frac{\chi_{i,j+1} \cdot c_{i,j}}{h_y} + \frac{\chi_{i+1,j} \cdot b_{i,j}}{h_x}$

A.2.8 Stencil $\mathcal{S}^{2,q,p}$

Stencil weight for $p_{i,j}$:
$2 \cdot b_{i,j}$

A.2.9 Stencil $\mathcal{S}^{2,q,q}$

Stencil weight for $q_{i,j}$:
$2 \cdot c_{i,j}$

Bibliography

[BA96] M. J. Black, P. Anandan. “The robust estimation of multiple motions: parametric and piecewise-smooth flow fields”. In: *Computer Vision and Image Understanding* 63.1 (1996), pp. 75–104. DOI: 10.1006/cviu.1996.0006 (cit. on p. 8).

[BBPW04] T. Brox, A. Bruhn, N. Papenberg, J. Weickert. “High accuracy optical flow estimation based on a theory for warping”. In: *Proc. European Conference on Computer Vision (ECCV)*. LNCS 3024. Springer, 2004, pp. 25–36. DOI: 10.1007/978-3-540-24673-2_3 (cit. on pp. 8, 9, 23).

[BDB13] J. Braux-Zin, R. Dupont, A. Bartoli. “A general dense image matching framework combining direct and feature-based costs”. In: *Proc. International Conference on Computer Vision (ICCV)*. 2013, pp. 185–192. DOI: 10.1109/ICCV.2013.30 (cit. on pp. 8, 9, 32).

[BKP10] K. Bredies, K. Kunisch, T. Pock. “Total Generalized Variation”. In: *SIAM Journal on Imaging Sciences* 3.3 (2010), pp. 492–526. DOI: 10.1137/090769521 (cit. on pp. 9, 32).

[Bru06] A. Bruhn. “Variationelle optische Flussberechnung: Präzise Modellierung und effiziente Numerik”. PhD thesis. Saarland University, 2006. DOI: 10.22028/D291-25874 (cit. on pp. 7, 8, 14, 23, 67).

[BSL+11] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, R. Szeliski. “A database and evaluation methodology for optical flow”. In: *International Journal of Computer Vision* 92.1 (2011), pp. 1–31. DOI: 10.1007/s11263-010-0390-2 (cit. on p. 65).

[BWSB12] D. J. Butler, J. Wulff, G. B. Stanley, M. J. Black. “A naturalistic open source movie for optical flow evaluation”. In: *Proc. European Conference on Computer Vision (ECCV)*. LNCS 7577. Springer, 2012, pp. 611–625. DOI: 10.1007/978-3-642-33783-3_44 (cit. on p. 65).

[CBAB97] P. Charbonnier, L. Blanc-Feraud, G. Aubert, M. Barlaud. “Deterministic edge-preserving regularization in computed imaging”. In: *IEEE Transactions on Image Processing* 6.2 (1997), pp. 298–311. DOI: 10.1109/83.551699 (cit. on p. 28).

[CMM00] T. Chan, A. Marquina, P. Mulet. “High-order Total Variation-based image restoration”. In: *SIAM Journal on Scientific Computing* 22.2 (2000), pp. 503–516. DOI: 10.1137/S1064827598344169 (cit. on p. 9).

[DSV+14] O. Demetz, M. Stoll, S. Volz, J. Weickert, A. Bruhn. “Learning brightness transfer functions for the joint recovery of illumination changes and optical flow”. In: *Proc. European Conference on Computer Vision (ECCV)*. LNCS 8689. Springer, 2014, pp. 455–471. DOI: 10.1007/978-3-319-10590-1_30 (cit. on pp. 8, 9, 31).

[GLU12] A. Geiger, P. Lenz, R. Urtasun. “Are we ready for autonomous driving? The KITTI vision benchmark suite”. In: *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012, pp. 3354–3361. DOI: 10.1109/CVPR.2012.6248074 (cit. on p. 65).

[GVWT13] P. Garrido, L. Valgaert, C. Wu, C. Theobalt. “Reconstructing detailed dynamic face geometry from monocular video”. In: *ACM Transactions on Graphics* 32.6 (2013). DOI: 10.1145/2508363.2508380 (cit. on p. 7).

[HO01] N. Hansen, A. Ostermeier. “Completely derandomized self-adaptation in evolution strategies”. In: *Evolutionary Computation* 9.2 (2001), pp. 159–195. DOI: 10.1162/106365601750190398 (cit. on pp. 63, 64).

[HS81] B. K. Horn, B. G. Schunck. “Determining optical flow”. In: *Proc. Techniques and Applications of Image Understanding*. Vol. 0281. International Society for Optics and Photonics. SPIE, 1981, pp. 319–331. DOI: 10.1117/12.965761 (cit. on pp. 8, 9, 13, 14, 63).

[HSW15] D. Hafner, C. Schroers, J. Weickert. “Introducing maximal anisotropy into second order coupling models”. In: *Proc. German Conference on Pattern Recognition (GCPR)*. LNCS 9358. Springer, 2015, pp. 79–90. DOI: 10.1007/978-3-319-24947-6_7 (cit. on pp. 9, 34).

[HWS+13] A. Hewer, J. Weickert, H. Seibert, T. Scheffer, S. Diebels. “Lagrangian strain tensor computation with higher order variational models”. In: *Proc. British Machine Vision Conference (BMVC)*. 2013, pp. 129.1–129.10. DOI: 10.5244/C.27.129 (cit. on pp. 8, 9).

[KC13] G. Kuschk, D. Cremers. “Fast and accurate large-scale stereo reconstruction using variational methods”. In: *Proc. International Conference on Computer Vision Workshops (ICCVW)*. 2013, pp. 700–707. DOI: 10.1109/ICCVW.2013.96 (cit. on pp. 8, 9).

[Kie18] H. Kielhöfer. *Calculus of Variations: An Introduction to the One-Dimensional Theory with Examples and Exercises*. Vol. 67. Springer, 2018. DOI: 10.1007/978-3-319-71123-2 (cit. on pp. 11, 12).

[KNPS68] J. Kačur, J. Nečas, J. Polák, J. Souček. “Convergence of a method for solving the magnetostatic field in nonlinear media”. In: *Aplikace Matematiky* 13.6 (1968), pp. 456–465 (cit. on p. 26).

[KPL11] Y.-C. Kuo, N.-S. Pai, Y.-F. Li. “Vision-based vehicle detection for a driver assistance system”. In: *Computers & Mathematics with Applications* 61.8 (2011), pp. 2096–2100. DOI: 10.1016/j.camwa.2010.08.081 (cit. on p. 7).

[KVR+09] J. Klappstein, T. Vaudrey, C. Rabe, A. Wedel, R. Klette. “Moving object segmentation using optical flow and depth information”. In: *Proc. Pacific-Rim Symposium on Image and Video Technology (PSIVT)*. LNCS 5414. Springer, 2009, pp. 611–623. DOI: 10.1007/978-3-540-92957-4_53 (cit. on p. 7).

[LBL13] F. Lenzen, F. Becker, J. Lellmann. “Adaptive second-order Total Variation: an approach aware of slope discontinuities”. In: *Proc. International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*. LNCS 7893. Springer, 2013, pp. 61–73. DOI: 10.1007/978-3-642-38267-3_6 (cit. on p. 9).

[LMB02] G. Lefaix, T. Marchand, P. Bouthemy. “Motion-based obstacle detection and tracking for car driving assistance”. In: *Proc. International Conference on Pattern Recognition (ICPR)*. Vol. 4. 2002, pp. 74–77. DOI: 10.1109/ICPR.2002.1047403 (cit. on p. 7).

[Mau19] D. Maurer. “Adaptive algorithms for 3D reconstruction and motion estimation”. PhD thesis. Stuttgart University, 2019. DOI: 10.18419/opus-10651 (cit. on p. 29).

[MB18] D. Maurer, A. Bruhn. “ProFlow: learning to predict optical flow”. In: *Proc. British Machine Vision Conference (BMVC)*. 2018 (cit. on pp. 10, 67).

[MG15] M. Menze, A. Geiger. “Object Scene Flow for Autonomous Vehicles”. In: *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3061–3070. DOI: 10.1109/CVPR.2015.7298925 (cit. on pp. 65, 67).

[MMGB18] D. Maurer, N. Marniok, B. Goldluecke, A. Bruhn. “Structure-from-motion-aware patchmatch for adaptive optical flow estimation”. In: *Proc. European Conference on Computer Vision (ECCV)*. LNCS 11212. Springer, 2018, pp. 575–592. DOI: 10.1007/978-3-030-01237-3_35 (cit. on p. 10).

[MSB17a] D. Maurer, M. Stoll, A. Bruhn. “Order-adaptive and illumination-aware variational optical flow refinement”. In: *Proc. British Machine Vision Conference (BMVC)*. 2017, pp. 150.1–150.13. DOI: 10.5244/C.31.150 (cit. on p. 10).

[MSB17b] D. Maurer, M. Stoll, A. Bruhn. “Order-adaptive regularisation for variational optical Flow: global, local and in between”. In: *Proc. International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*. LNCS 10302. Springer, 2017, pp. 550–562. DOI: 10.1007/978-3-319-58771-4_44 (cit. on pp. 3, 8–10, 33–37, 55–57, 61, 63, 65–70, 75).

[MSB18] D. Maurer, M. Stoll, A. Bruhn. “Directional Priors for Multi-Frame Optical Flow.” In: *Proc. British Machine Vision Conference (BMVC)*. 2018 (cit. on p. 10).

[MSV+17] D. Maurer, M. Stoll, S. Volz, P. Gairing, A. Bruhn. “A comparison of isotropic and anisotropic second order regularisers for optical flow”. In: *Proc. International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*. LNCS 10302. Springer, 2017, pp. 537–549. DOI: 10.1007/978-3-319-58771-4_43 (cit. on pp. 8, 9, 34).

[NE86] H. Nagel, W. Enkelmann. “An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8.5 (1986), pp. 565–593. DOI: 10.1109/TPAMI.1986.4767833 (cit. on pp. 8, 9).

[NM65] J. A. Nelder, R. Mead. “A simplex method for function minimization”. In: *The Computer Journal* 7.4 (1965), pp. 308–313. DOI: 10.1093/comjnl/7.4.308 (cit. on pp. 63, 64).

[OS14] N. Onkarappa, A. D. Sappa. “Speed and texture: an empirical study on optical-flow accuracy in ADAS scenarios”. In: *IEEE Transactions on Intelligent Transportation Systems* 15.1 (2014), pp. 136–147. DOI: 10.1109/TITS.2013.2274760 (cit. on p. 7).

[PM90] P. Perona, J. Malik. “Scale-space and edge detection using anisotropic diffusion”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.7 (1990), pp. 629–639. DOI: 10.1109/34.56205 (cit. on p. 29).

[RBP14] R. Ranftl, K. Bredies, T. Pock. “Non-local Total Generalized Variation for optical flow estimation”. In: *Proc. European Conference on Computer Vision (ECCV)*. LNCS 8689. Springer, 2014, pp. 439–454. DOI: 10.1007/978-3-319-10590-1_29 (cit. on p. 9).

[RRBW12] L. L. Rakêt, L. Roholm, A. Bruhn, J. Weickert. “Motion compensated frame interpolation with a symmetric optical flow constraint”. In: *Proc. International Symposium on Visual Computing (ISVC)*. LNCS 7431. Springer, 2012, pp. 447–457. DOI: 10.1007/978-3-642-33179-4_43 (cit. on p. 7).

[Saa03] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Second Edition. Society for Industrial and Applied Mathematics, 2003. DOI: 10.1137/1.9780898718003 (cit. on pp. 19–21).

[SBK10] N. Sundaram, T. Brox, K. Keutzer. “Dense point trajectories by GPU-accelerated large displacement optical flow”. In: *Proc. European Conference on Computer Vision (ECCV)*. LNCS 6311. Springer, 2010, pp. 438–451. DOI: 10.1007/978-3-642-15549-9_32 (cit. on pp. 7, 17).

[SP01] Soo-Chul Han, C. I. Podilchuk. “Video compression with dense motion fields”. In: *IEEE Transactions on Image Processing* 10.11 (2001), pp. 1605–1612. DOI: 10.1109/83.967388 (cit. on p. 7).

[SVMB17] M. Stoll, S. Volz, D. Maurer, A. Bruhn. “A time-efficient optimisation framework for parameters of optical flow methods”. In: *Proc. Scandinavian Conference on Image Analysis (SCIA)*. LNCS 10269. Springer, 2017, pp. 41–53. DOI: 10.1007/978-3-319-59126-1_4 (cit. on pp. 63–65).

[TPCB08] W. Trobin, T. Pock, D. Cremers, H. Bischof. “An unbiased second-order prior for high-accuracy motion estimation”. In: *Proc. German Conference on Pattern Recognition (GCPR)*. LNCS 5096. Springer, 2008, pp. 396–405. DOI: 10.1007/978-3-540-69321-5_40 (cit. on p. 8).

[Wei98] J. Weickert. *Anisotropic Diffusion in Image Processing*. Vol. 1. Teubner Stuttgart, 1998 (cit. on p. 48).

[WS01] J. Weickert, C. Schnörr. “A theoretical framework for convex regularizers in PDE-based computation of image motion”. In: *International Journal of Computer Vision* 45.3 (2001), pp. 245–264. DOI: 10.1023/A:1013614317973 (cit. on p. 14).

[WTP+09] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, H. Bischof. “Anisotropic huber-L1 optical flow”. In: *Proc. British Machine Vision Conference (BMVC)*. 2009, pp. 108.1–108.11. DOI: 10.5244/C.23.108 (cit. on pp. 8, 9).

[WWW13] J. Weickert, M. Welk, M. Wickert. “L2-stable nonstandard finite differences for anisotropic diffusion”. In: *Proc. International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*. LNCS 7893. Springer, 2013, pp. 380–391. DOI: 10.1007/978-3-642-38267-3_32 (cit. on pp. 48, 49).

[You71] D. M. Young. *Iterative Solution of Large Linear Systems*. Academic Press New York, 1971. DOI: 10.1016/C2013-0-11733-3 (cit. on p. 20).

[ZBW11] H. Zimmer, A. Bruhn, J. Weickert. “Optic flow in harmony”. In: *International Journal of Computer Vision* 93.3 (2011), pp. 368–388. DOI: 10.1007/s11263-011-0422-6 (cit. on pp. 8, 9, 30, 31).

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Stuttgart, 19.03.2020 L. Mehl

Ort, Datum, Unterschrift