Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

Taxonomy of First and Second Order Regularizers for Variational Motion Estimation

Patrick Gairing

Course of Study:	Informatik
Examiner:	Prof. Dr.-Ing. Andrés Bruhn
Supervisor:	Prof. Dr.-Ing. Andrés Bruhn
Commenced:	October 10, 2016
Completed:	April 11, 2017
CR-Classification:	G.1.6, I.2.10, I.4.8

Abstract

Automatic detection of motion in images is one of the main challenges in the field of computer vision. It can be used in many applications to improve their results. One essential part of motion detection is the computation of a displacement vector field (optical flow) between consecutive image pairs. Techniques that allow the most accurate results at the time belong to global optimization methods (variational methods). Using variational methods one typically has to minimize energy (cost) functionals.

In this thesis we model different energy functionals and study as well as evaluate them in terms of their accuracy with respect to motion estimation based on common test sequences. An energy functional penalizes deviations of model assumptions. It consists of a data and a regularization term (smoothness term). While the data term with its constancy assumptions is used to detect corresponding structures between different images, the regularization term allows a smoothing of the motion field to provide a dense solution. We focus on modeling different regularizers in this work, including such ones, which were successfully used in variational methods for other fields but not considered for motion estimation until now.

The thesis is divided into three main parts: (1) We initially present a prototypical variational method to estimate optical flow. Starting with the continuous model, we discretize the method and a possible solution of the resulting equations is derived. At this point, more advanced data terms are introduced as well. (2) Afterwards, we present regularizers of different order and combination. Furthermore, we consider an isotropic and an anisotropic version of each regularizer. Incorporating additional image information we expect the direction-dependent smoothing of anisotropic regularizers to provide more accurate solutions. (3) Finally, we compare the different regularizers to each other, where we also make use of the several data terms.

Kurzfassung

Die automatisierte Erkennung von Bewegung in Bildern ist, vor allem wegen der Nützlichkeit in zahlreichen Anwendungsgebieten, eine der Hauptherausforderungen auf dem Gebiet des Maschinensehens. Ein grundlegender Bestandteil der Bewegungserkennung ist die Berechnung eines Verschiebungsvektorfelds (optischer Fluss) zwischen aufeinanderfolgenden Bildpaaren. Verfahren, die zurzeit die genaueste Berechnung des optischen Flusses ermöglichen, gehören zu den globalen Optimierungsmethoden (Variationsansätze). In solchen Variationsansätzen werden häufig Energiefunktionale (Kostenfunktionale) minimiert.

In dieser Masterarbeit werden verschiedene Kostenfunktionale modelliert und auf Basis gemeinsamer Testsequenzen hinsichtlich ihrer Genauigkeit der Bewegungsschätzung untersucht und evaluiert. Ein Kostenfunktional bestraft Abweichungen von Annahmen der Modellierung. Es besteht typischerweise aus einem Daten- und einem Regularisierungsterm (Glattheitsterm). Während der Datenterm mit Konstanzannahmen zur Erkennung korrespondierender Strukturen in unterschiedlichen Bildern genutzt wird, erlaubt der Regularisierungsterm eine Glättung des Bewegungsfelds, um eine dichte Lösung zu liefern. Der Schwerpunkt der Modellierung in dieser Arbeit wird auf verschiedene Regularisierer gelegt - unter anderem auch auf solche, die bisher erfolgreich in Varationsansätzen für andere Bereiche wie dem Entrauschen oder Focus Fusion, aber noch nicht zur Bewegungsschätzung genutzt wurden.

Die Masterarbeit lässt sich in drei Hauptabschnitte unterteilen: (1) Zunächst wird ein variationelles Verfahren zur Schätzung des optischen Flusses vorgestellt. Vom kontinuierlichen Modell ausgehend wird diskretisiert und eine mögliche Lösung des Verfahrens hergeleitet. An dieser Stelle werden auch schon weiterentwickelte Datenterme eingeführt. (2) Anschließend werden Regularisierer unterschiedlichen Grades und Kombination präsentiert. Je Regularisierer wird zudem eine isotrope und eine anisotrope Variante betrachtet. Unter Berücksichtigung zusätzlicher Bildinformationen soll die richtungsgesteuerte Glättung der anisotropen Varianten eine genauere Schätzung liefern. (3) Zum Schluss werden die verschiedenen Regularisierer in Verbindung mit den unterschiedlichen Datentermen miteinander verglichen.

Acknowledgments

This master thesis would not have been possible without the aid and support from many other people. They all deserve special mention here.

First of all, i would like to thank Professor *Andrés Bruhn* for offering me an interesting topic to work with and supervising my master thesis. His lectures awaken my interests to immerse myself in the field of computer vision. In this context, i also want to thank *Michael Stoll*. He provided explanations about the optical flow framework and helped me to solve occurring problems regarding it. I also want to express my gratitude to both of them, since they gave me the opportunity to work in the Intelligent Systems group as a tutor.

Moreover, i want to thank my fellow student *Tilo Pfannkuch*. It was a pleasure to study side by side with him during our time at the university. Further thanks go to my long-time friends *Christian Schulz* and *Maximilian Schnorr*, who i know since our early days of childhood. They always helped me with assistance and advice. I also want to thank *Christian Schulz* for proofreading parts of my thesis.

Finally, i would like to express my gratitude to those people who are most important to me: First, I want to thank my girlfriend *Eunyoung Bae* for her love, constant support and for introducing me to an entirely new side of life. And second, i would like to thank my parents *Mathias* and *Lilli Gairing* for encouraging me to begin and finish studying and always helping me trough difficult times.

Thank you all!

Contents

1 Introduction **9**

1.1 General Variational Optical Flow Model . . . 11
1.2 Related Work and Our Contribution . . . 13
1.3 Organization . . . 15

2 Fundamentals **17**

2.1 Images . . . 17
2.2 Norms . . . 18
2.3 Derivatives . . . 19
2.4 Calculus of Variations . . . 21
2.5 Quality Measures . . . 23

3 Optical Flow **25**

3.1 Data Term . . . 25
3.1.1 Gray Value Constancy . . . 26
3.1.2 Gradient Constancy . . . 27
3.1.3 Complete Rank Transform . . . 27
3.1.4 Color Images . . . 27
3.1.5 Normalization . . . 29
3.1.6 Robustification . . . 30
3.1.7 Quantitative Results . . . 31
3.2 Smoothness Term . . . 34
3.2.1 Generic Form and Relation to Diffusion . . . 35
3.2.2 Robustification . . . 35
3.3 Minimization . . . 38
3.3.1 Fixed Point Iteration . . . 41
3.3.2 Incremental Computation . . . 41
3.3.3 Coarse-To-Fine Strategy . . . 43
3.4 Discretization . . . 44
3.5 Solving the System of Equations . . . 46

4 Regularizers in Detail **49**

4.1 First Order . . . 50
4.1.1 Isotropic . . . 50
4.1.2 Anisotropic . . . 51
4.2 Second Order . . . 52
4.2.1 Isotropic . . . 52
4.2.2 Anisotropic . . . 53

4.3 First and Second Order Combined 54
4.3.1 Isotropic 54
4.3.2 Anisotropic 56
4.4 Coupled Regularization 58
4.4.1 Isotropic 58
4.4.2 Anisotropic 60
4.5 Summary 62

5 Experimental Evaluation **65**
5.1 Data Sets 66
5.1.1 Middlebury Benchmark 66
5.1.2 Sintel Benchmark 69
5.1.3 KITTI Benchmark 71
5.2 Comparison of the Regularizers 72
5.2.1 Results - Middlebury 73
5.2.2 Results - Sintel 79
5.2.3 Results - KITTI 86
5.3 Summary 95

6 Conclusion **97**

A Regularization Stencils **99**
A.1 First Order Isotropic 99
A.2 First Order Anisotropic 99
A.3 Second Order Isotropic 100
A.4 Second Order Anisotropic 101
A.5 Coupled Regularization Isotropic 102
A.6 Coupled Regularization Anisotropic 103

B Derivations **105**
B.1 Second Order Isotropic Discretization 105
B.2 Second Order Anisotropic Discretization 106
B.3 Coupled Isotropic Discretization 109
B.4 Coupled Anisotropic Discretization 110

Bibliography **111**

1 Introduction

Computer vision is a large field that is mainly concerned with image analysis and pattern recognition. When it comes to image analysis, image sequences and estimating motion within these sequences play an important role. To estimate motion one has to compute a displacement vector field, which is known as *optical flow* in common literature. One of the main issues of optical flow computation is to find corresponding structures between two consecutive images. Such so-called *correspondence problems* are not only to be solved for motion estimation, but also in other areas such as depth reconstruction or medical image registration. Hence, we can say solving correspondence problems is very fundamental in computer vision. Unfortunately, finding correspondences is not a trivial process.

Regarding optical flow estimation we distinguish between *local differential* and *global differential* methods to establish correspondences. Local methods incorporate constancy assumptions that should apply to neighborhoods of certain size around pixels. In the next step they minimize energy functions for each pixel locally by calculating constant parameters that fit best to the model constraints within these neighborhoods. One major problem that can occur here is that unique solutions do not always exist due to the nature of images (*aperture problem*). In fact, we can compute only the part of the flow (*normal flow*) that is orthogonal to image edges. Local methods try to tackle this problem by using the already mentioned neighborhood, but there are still some cases where the flow cannot be computed accurately. To overcome these situations, global methods compute a minimizer of a global energy functional on the entire image domain. In contrast to local methods, we do not minimize pixelwise but consider a set of functions as a minimizer and therefore determine all parts of the flow simultaneously (*variational methods*). To do so, a second assumption supplements the constancy assumption. It is known as the smoothness constraint and it aims at regularizing the resulting optical flow. Thus, variational methods provide a unique solution to a previous ill-posed problem.

As its name implies, the smoothness constraint in its basic variant favors motion fields with only small spatial variations (small variations of flow derivatives) and hence smoothes/regularizes the entire flow. But regularizers can be modeled in many ways depending on the image and expected flow. For example, we can modify the regularizer with respect to flow discontinuities by using *subquadratic penalizer* functions and thus better differentiate between the motion of various objects in the image. We can use different orders of derivatives to model different kinds of motion and even take directional constraints into account to allow direction-dependent smoothing (*anisotropy*). Several combinations of these modifications are also possible to further improve optical flow estimation. We look at them in detail later in this work.

Fields of Application. Literature shows that optical flow estimation and the resulting motion field can be successfully used in various areas ranging from image processing over tracking and navigation tasks to automated driver assistance systems. If we consider image processing for example, optical flow is used for *compression* of video sequences for a long time. On the one hand, it can be used to actively reduce the amount of necessary space to store videos. Here the basic concept is to divide a video in key frames (directly stored images using e.g. JPEG) and store remaining frames as flow fields indicating differences between frames [1]. On the other hand, further development of optical flow methods [2] made it possible to speed up encoding time of High Efficiency Video Coding (HEVC), the current state-of-the-art video coding standard [3]. Optical flow helps to decide where to skip certain parts of the HEVC pipeline. Another prominent field of application is *robot navigation*, which is not done manually but acts fully autonomously. Optical flow is used here to determine characteristics about an a priori unknown environment such as obstacle disposition, time to contact between robot and obstacles, and depth. Robots then utilize this knowledge to safely navigate through the environment by avoiding obstacles [4]. Combined with infrared cameras optical flow also allows to characterize obstacles as humans. Thus, it provides support for surveillance tasks [5]. Detecting obstacles, to mention another practical application of optical flow, is also of particular interest for *driver assistance systems*, which may help a car driver to avoid potential accidents [6] (see Figure 1.1).

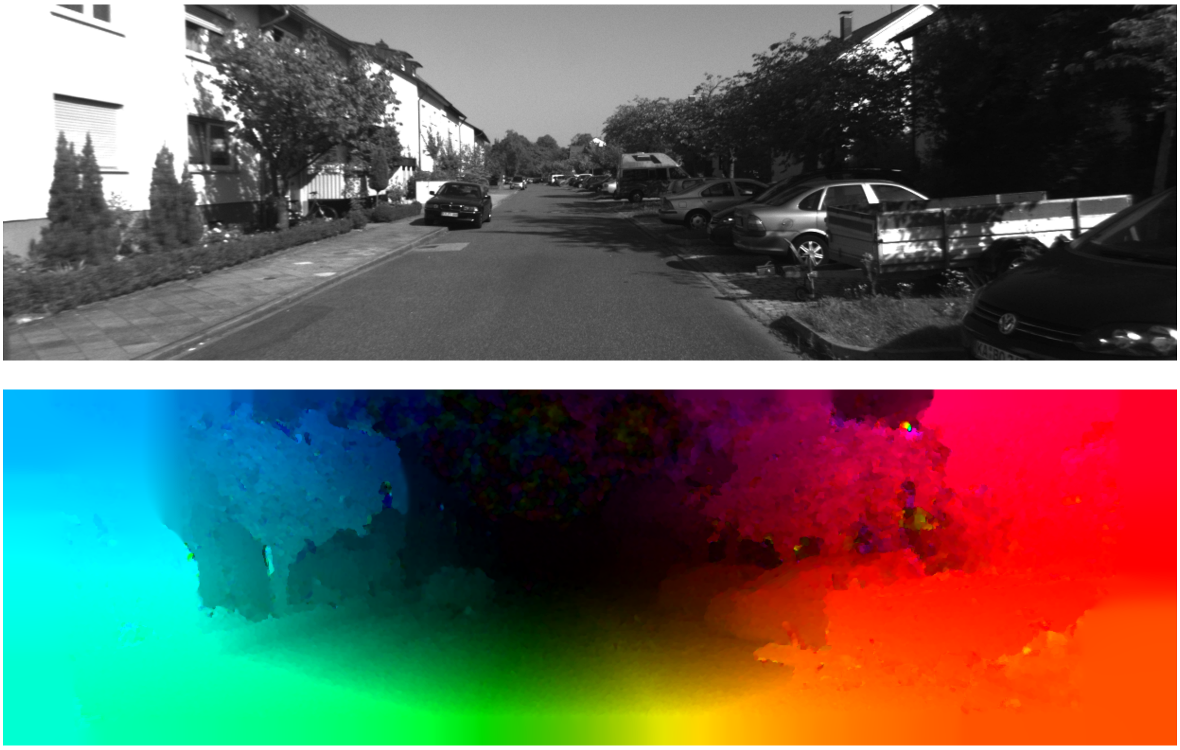

Figure 1.1: Possible application in a driver assistance system. **Top:** First image of one input image pair taken from the KITTI benchmark data set [7]. **Bottom:** Computed optical flow, where colors indicate the direction of motion.

1.1 General Variational Optical Flow Model

Before we have an introductory look on variational methods, we briefly discuss alternative approaches [8] to find corresponding structures and thus estimate motion fields. There are *area-based techniques*, which consider local image information in a neighborhood specified by a block around a reference pixel in one image and compare them to some candidate blocks in another image [9, 10]. On the contrary, instead of computing a dense motion field, *feature-based approaches* only select a few but distinctive image features and solve correspondence problems between them [11, 12]. Finally, we have the already mentioned *local differential methods* [13], which minimize local energy functions instead of global functionals. Both local and variational methods share the same idea how to include constancy constraints, but to overcome the aperture problem variational methods use additional smoothness constraints enforcing global smoothness of the optical flow.
One first approach regarding regularizers was made in the prominent work of Horn and Schunck [14]. We consider $f(\mathbf{x})$ as a scalar-valued *image sequence* with $\mathbf{x} := (x, y, t)^\top$, where $(x, y)^\top \in \Omega$ denotes a specific location within a continuous rectangular image domain $\Omega \subset \mathbb{R}^2$ and $t \geq 0$ denotes time. In a discrete sense t specifies a certain image of an image sequence. We then want to estimate the optical flow field $\mathbf{w} := (u, v, 1)^\top$, where $u(\mathbf{x})$ and $v(\mathbf{x})$, respectively, describe the displacements in x- and y-direction and 1 identifies the subsequent image $t + 1$ (see Figure 1.2), by minimizing a suitable global energy functional that can be written in its most general form as

$$E(u, v) = \int_\Omega \underbrace{D(u, v)}_{\text{Data term}} + \alpha \underbrace{R(\nabla_2 u, \nabla_2 v)}_{\text{Regularization term}} \mathrm{d}x\mathrm{d}y \tag{1.1}$$

with the spatial gradient $\nabla_2 := (\partial_x, \partial_y)^\top$. The optical flow field acts as the minimizing function and the regularization parameter α helps to balance both terms. The data term typically contains constancy assumptions, whereas the regularizer contains smoothness assumptions. Both terms penalize deviations from these assumptions.

The most frequently used constraint in the data term is the so-called *brightness constancy assumption* (BCA). It states that intensity values of image structures remain the same after

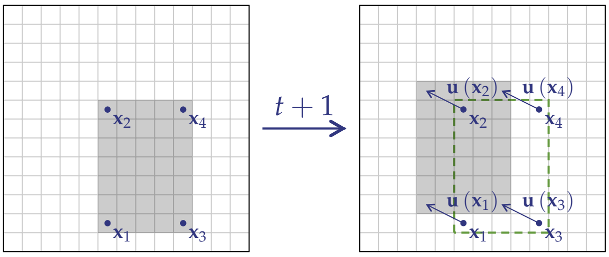
$\mathbf{x}_2$
$\mathbf{x}_4$
$\mathbf{x}_1$
$\mathbf{x}_3$
$t+1$
$\mathbf{u}(\mathbf{x}_2)$
$\mathbf{u}(\mathbf{x}_4)$
$\mathbf{u}(\mathbf{x}_1)$
$\mathbf{u}(\mathbf{x}_3)$

Figure 1.2: Displacement vector field $\mathbf{u} := (u, v)^\top$ for corresponding pixels at position $\mathbf{x}$ between two images at time t and $t + 1$. **Left:** Image t. **Right:** Image $t + 1$.

motion. We write it as

$$f\left(x,y,t\right)=f\left(x+u,y+v,t+1\right). \tag{1.2}$$

If we assume the motion to be small and f is sufficiently smooth and differentiable, we can perform a first order Taylor expansion around a point $\mathbf{x}$ to linearize the BCA

$$f\left(x+u,y+v,t+1\right)\approx f\left(x,y,t\right)+\frac{u^1}{1}f_x+\frac{v^1}{1}f_y+\frac{1^1}{1}f_t \tag{1.3}$$

where the abbreviations f_x, f_y and f_t denote partial derivatives with respect to x, y and t. The linearization is done to manage the implicitness and nonlinearity of u and v in Equation 1.2. Approximation 1.3 applied to Equation 1.2 yields the *linearized optical flow constraint* (OFC)

$$f_xu+f_yv+f_t=\nabla_3f^\top\mathbf{w}=0 \tag{1.4}$$

with $\nabla_3:=\left(\partial_x,\partial_y,\partial_t\right)^\top$ as spatiotemporal gradient operator. That way, our data term reads

$$D\left(u,v\right):=\left(\nabla_3f^\top\mathbf{w}\right)^2. \tag{1.5}$$

Since we penalize all deviations from zero (positive and negative values), the constraint gets squared. As we can see in Equation 1.4 there are two unknowns u and v but only one single equation, hence infinitely many solutions for the flow components exist. This is the aforementioned aperture problem. Without additional constraints the OFC only allows to compute flow components parallel to the spatial image gradient ∇_2f. These components are called *normal flow* [15] which is defined as

$$\mathbf{w}_n:=\left(\mathbf{u}_n^\top,1\right)^\top:=\left(-\frac{f_t}{\left|\nabla_2f\right|}\frac{\nabla_2f^\top}{\left|\nabla_2f\right|},1\right)^\top. \tag{1.6}$$

To tackle the aperture problem a second constraint is introduced in our variational model: the smoothness constraint. It regularizes the flow field at locations where the data term does not contribute to the solution ($\left|\nabla_2f\right|\approx0$). A *filling-in effect* occurs at such locations taking neighborhood information into account and adapting the local flow to fulfill the constraint in the best way possible. This effect is closely related to an implicit diffusion process to achieve dense flow fields. Mathematically, this is realized by penalizing deviations from the squared magnitude of the spatial flow gradient. The resulting smoothness term reads [14]

$$R\left(\nabla_2u,\nabla_2v\right):=\left|\nabla_2u\right|^2+\left|\nabla_2v\right|^2. \tag{1.7}$$

This constraint ends up in a homogeneous diffusion that smoothes the entire flow field the same way without adapting to image or flow data.

Data and smoothness term applied to our general variational optical flow model in Equation 1.1 yields the Horn and Schunck energy functional

$$E\left(u,v\right)=\int_\Omega\left(\nabla_3f^\top\mathbf{w}\right)^2+\alpha\left(\left|\nabla_2u\right|^2+\left|\nabla_2v\right|^2\right)\mathrm{d}x\mathrm{d}y. \tag{1.8}$$

Later in this work we will discuss many modifications of the data and the smoothness term, especially of the latter. Instead of ∇_2 we will often just use the shorter notation ∇.

1.2 Related Work and Our Contribution

Both the data and the smoothness term of variational methods and their interaction are subject to many studies in research. It is often the case that literature focuses on how to design either the data or the smoothness term. Therefore, it makes sense to consider them here separately as well.

Data Term. Regarding design of data terms, one is interested in modifying constraints in such a way that qualitative performance is increased under varying conditions of a captured scene like illumination changes, which can be very challenging to face. One simple extension to the data term in Equation 1.5 that is proposed by Uras et al. [16] is to consider constancy of the spatial brightness gradient $\nabla_2 f$ instead of constant brightness f. Contrary to the BCA we obtain two equations for two unknowns this time, which may allow to overcome the aperture problem. Papenberg et al. [17] even examine higher order derivatives to be used in the data term. However, one cannot automatically assume the higher the derivative, the better is qualitative performance of motion estimation. Instead of utilizing higher order derivatives to achieve invariance under illumination changes, Demetz et al. [18] suggest novel descriptors based on the order of gray values in a neighborhood. Despite being relatively easy and intuitive to understand, this approach yields good results in experiments. Due to the linearization we mentioned in Section 1.1 a *strictly convex* energy functional is obtained. That means, each minimization strategy that is globally convergent can find a unique solution since a global minimum exists. However, linearization is only valid for small displacements. In image sequences with very fast motion or coarse temporal sampling this condition is violated. It is possible to postpone the linearization to the numerical scheme, but in this case we have to minimize a *nonconvex* energy functional with multiple local minima, that may cause algorithms to get trapped there. Brox et al. [2] alleviate this problem by presenting a multiscale coarse-to-fine warping approach. To determine a global minimum they start to work on coarse versions of the input images and then successively compute minima on each coarse-to-fine hierarchy level until the original problem is solved.
Since we penalize deviations from constancy assumptions in a quadratic way (see Equation 1.5), outliers caused by noise and occlusions get much higher influence on the solution than they originally should. To reduce their influence a subquadratic penalizer function such as the regularized L_1 norm is applied to the data term by Brox et al. [2]. Originally, the idea to penalize deviations less severely comes from robust statistics as discussed by Black and Anandan [19]. Furthermore, if multiple constancy assumptions are used within the data term and they can be fulfilled independently, Bruhn and Weickert [20] use a separate penalization (*robustification*). Unfortunately, these penalizer functions require the minimization to solve a nonlinear system of equations which is mathematically and computationally more challenging.

Smoothness Term. Similar to data terms we can apply subquadratic penalizers to the regularizer. While the quadratic term in Equation 1.7 performs a homogeneous diffusion, subquadratic penalizers allow piecewise smooth flow fields and hence adaptively smooth

flow discontinuities, which usually occur at object boundaries. As in the case of data terms one possible choice is the regularized *total variation* (TV) regularization. It was first used in context of noise removal by Rudin et al. [21]. First order derivatives allow to model edges in the solution but also produce staircasing effects (oversegmentation in constant stairs) with TV regularization. To reduce such artifacts Chambolle and Lions [22] use direct first and second order derivative regularization in an *infimal convolution* (INF) of energy functionals. Second order derivatives favor affine motion (Demetz et al. [23]), i.e. allow kinks in the solution. Trobin et al. [24] employ only deviations from second order derivatives as a prior to regularize the flow field. Since second order derivatives are not orthogonal, they use a new operator to set up a norm that prevents a biased behavior of preferring some certain affine flows. A more generalized variant of TV regularization, the so-called *total generalized variation* (TGV), proposed by Bredies et al. [25] also involves higher order derivatives. TGV2, a second order case of TGV, penalizes first and second order derivatives as INF, but contains a coupling term that gives indirect access to second order derivatives. Such indirect, coupled approaches model discontinuities for each order explicitly.
All of the introduced smoothness terms so far do not incorporate directional information and smoothing only depends on the location. Such regularizers that treat diffusion in all directions equally are called *isotropic*. If methods steer diffusion across and along edges differently, they are said to be *anisotropic*. Zimmer et al. [15] present an anisotropic extension to TV regularization by integrating directions to the smoothness term. Similarly, Hafner et al. [26] modify TGV2 to regularize the solution in an anisotropic way. Another anisotropic second order TV approach is proposed by Lenzen et al. [27]. Their regularization depends on a locally adaptive change of regularization parameters depending on image location.

Interaction. The connection between data and smoothness term and their interplay is subject to the work proposed by Zimmer et al. [28]. Usually most data terms constrain the flow in one direction (normal flow). Homogeneous and isotropic diffusion approaches, however, also smooth in direction of data constraints. Anisotropic methods that utilize directional information taken from the *structure tensor* [29] smooth along image edges but keep discontinuities across them. If the data term contains the BCA, the data term and the anisotropic version of the Horn and Schunck smoothness term complement each other well. Since the structure tensor considers brightness edges only, advanced data terms like gradient constancy are not consistent with the directional information, though. The authors suggest to use directions derived from the *motion tensor* instead, i.e. from the constancy assumptions in the data term. Doing this, *constraint edges* and not images edges influence the regularization process.

Our Contribution. The aim of this thesis is to compare optical flow estimation quality of variational methods with various advanced data and smoothness terms on a common set of suitable image sequences. In this context we will consider higher order direct/coupled isotropic/anisotropic regularizers as introduced before. Since some of these regularizers were applied for other tasks, we need to properly adjust them for our needs. Therefore, the isotropic optical flow approaches and their anisotropic counterparts are studied and implemented based on the framework of the department, followed by an in-depth evaluation.

1.3 Organization

This thesis is organized as follows: Chapter 2 provides some theoretical foundations about images, notation and minimization. Furthermore, a metric is introduced to evaluate the flow estimations of the several methods. Chapter 3 is about optical flow and the variational method in general. We learn about the different data terms, a basic smoothness term, how to minimize and discretize them and, eventually, solve the resulting linear system of equations. After presenting the different smoothness terms in detail in Chapter 4, we show some experiments, evaluate the flow estimations and compare the different approaches to each other in Chapter 5. Finally, the thesis ends in Chapter 6 with a conclusion and an outlook on future work.

2 Fundamentals

Before we go into detail with variational optical flow methods, we first need to present some foundations and mathematical concepts. In Section 1.1 functions representing images are already mentioned but in addition to a formal definition we cover some discretization aspects in Section 2.1. Throughout this thesis some notation is used - its meaning is explained next in Section 2.2. We then continue with what needs to be considered when minimizing energy functionals and how the resulting equations can be solved in a discrete way. Therefore, we specify possible discretizations of derivatives in Section 2.3. Section 2.4 is about deriving the final system of equations based on calculus of variations. Eventually, we introduce some suitable evaluation metrics to measure the quality of our flow estimations in Section 2.5.

2.1 Images

Continuous. In our theoretical context we often consider images, also called *frames*, as continuous functions. They assign values of the co-domain $\mathbb{R}$ to various locations from the image domain Ω. Let $\Omega \subset \mathbb{R}^2$ be a rectangular image domain, then such image function f reads

$$f : \Omega \to \mathbb{R}. \tag{2.1}$$

In this case the co-domain specifies scalar values, e.g. gray values. In order to work with color images one has to extend Function 2.1 to a vector-valued function, i.e.

$$\mathbf{f} : \Omega \to \mathbb{R}^n \tag{2.2}$$

with $n \in \mathbb{N}$ representing the number of image channels. For example, if one uses the HSV or the RGB color representation, $n = 3$ applies.

Discrete. In order to use functions as frames in practice the image domain needs to discretized. We call this step *sampling*. Using step sizes $\mathbf{h} := (h_x, h_y)^\top \in \mathbb{N}^2$ in an equidistant, rectangular grid of size $m \cdot n$, we can discretize an image position $(x, y)^\top$ with

$$\left[\begin{pmatrix} x \\ y \end{pmatrix}\right]_{i,j} = \begin{pmatrix} i \cdot h_x \\ j \cdot h_y \end{pmatrix}. \tag{2.3}$$

Thus, for each index pair (i, j) we get $m \cdot n$ possible grid cells (*pixel* positions) in the discretized image domain

$$[\Omega] = \{1, \ldots, m\} \times \{1, \ldots, n\}. \tag{2.4}$$

Usually, pixels are quadratic, of same size and have the same distance to each other. For this reason $h_x = 1$ and $h_y = 1$ applies. Now we can evaluate an image f at the discretized locations by

$$[f]_{i,j} = f_{i,j} = f\left([(x,y)]_{i,j}\right). \tag{2.5}$$

Not only the image domain needs to be discretized, but also the co-domain. During this *quantization* a certain amount of storage is assigned to each pixel. For RGB images 24bits are used frequently. Each pixel of one channel is then mapped to

$$[f^c]_{i,j} \in \{0, \ldots, 255\} \tag{2.6}$$

where c denotes a certain color channel. Figure 2.1 shows an RGB image and its channel representations.

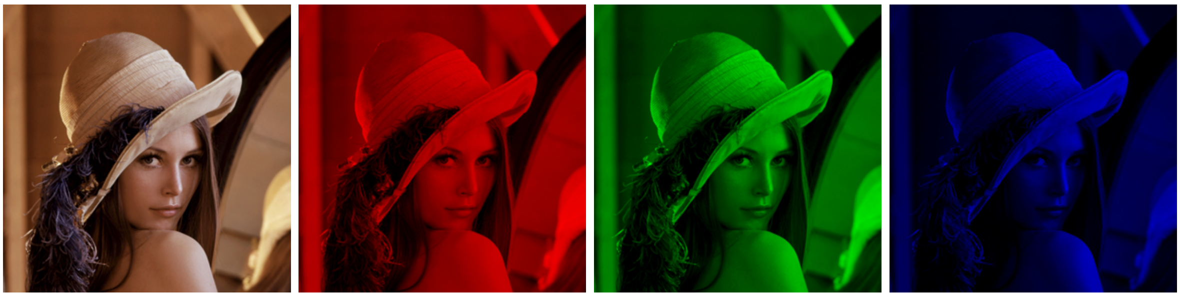

Figure 2.1: RGB image (Lena) and its separate color channels. **From Left to Right:** Full image, red channel, green channel and blue channel.

2.2 Norms

This section is about different notation for vector and matrix norms.

***p*-Norm.** A p-norm of a vector $\mathbf{a} \in \mathbb{R}^k$ is defined by

$$\|\mathbf{a}\|_p := \left(\sum_{i=1}^{k} |a_i|^p\right)^{\frac{1}{p}} \tag{2.7}$$

where $p \geq 1$. A special case of this norm that we use in this thesis often is the *Euclidean norm* with $p = 2$. We abbreviate it with

$$\|\mathbf{a}\|_2 = |\mathbf{a}| = \sqrt{\sum_{i=1}^{k} (a_i)^2}. \tag{2.8}$$

Matrix norm. We can extend p-norms to define matrix norms by considering all entries $a_{i,j}$ of a matrix $A \in \mathbb{R}^{k \times l}$ as entries of a vector with equal length. Such a norm is then defined as

$$\|A\|_p := \left(\sum_{i=1}^{k} \sum_{j=1}^{l} |a_{i,j}|^p \right)^{\frac{1}{p}}. \tag{2.9}$$

Again, a special case is given for $p = 2$. It is called *Frobenius norm*:

$$\|A\|_2 = \|A\|_{\mathrm{F}} = \sqrt{\sum_{i=1}^{k} \sum_{j=1}^{l} \left(a_{i,j}\right)^2}. \tag{2.10}$$

2.3 Derivatives

Derivatives play a major role in variational methods. Hence, we cover notations, their meaning and possible ways to discretize them in this section.

Partial Derivatives. Given a multivariate scalar-valued function $g : \mathbb{R}^n \to \mathbb{R}$, the *gradient* describes all possible *partial derivatives* with respect to the function's variables $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ as a vector:

$$\nabla g(\mathbf{x}) := \left(\frac{\partial g(\mathbf{x})}{\partial x_i} \right)^{\top}_{i=1,\ldots,n} = \begin{pmatrix} \frac{\partial g(\mathbf{x})}{\partial x_1} \\ \vdots \\ \frac{\partial g(\mathbf{x})}{\partial x_n} \end{pmatrix}. \tag{2.11}$$

It points in the direction of the steepest ascend. Applying the gradient to a scalar-valued function yields a vector-valued function $\nabla g := \mathbf{h} : \mathbb{R}^n \to \mathbb{R}^n$. The *divergence* of such a function $\mathbf{h}$ is defined by

$$\operatorname{div}(\mathbf{h}(\mathbf{x})) := \sum_{i=1}^{n} \frac{\partial h_i(\mathbf{x})}{\partial x_i} = \partial_{x_1} h_1 + \cdots + \partial_{x_n} h_n. \tag{2.12}$$

Since we have a special case of vector fields with $\mathbf{h} = \nabla g$, Equation 2.12 also defines the *Laplacian*:

$$\triangle g(\mathbf{x}) := \operatorname{div}(\nabla g(\mathbf{x})) = \sum_{i=1}^{n} \frac{\partial^2 h_i(\mathbf{x})}{\partial x_i \partial x_i}. \tag{2.13}$$

Jacobian. Gradients of multivariate vector-valued functions $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^m$ give the *Jacobian*

$$\mathcal{J}_{\mathbf{g}}(\mathbf{x}) := \left(\frac{\partial g_j(\mathbf{x})}{\partial x_i} \right)_{i=1,\ldots,n; j=1,\ldots,m} = \begin{pmatrix} \nabla g_1(\mathbf{x})^{\top} \\ \vdots \\ \nabla g_m(\mathbf{x})^{\top} \end{pmatrix} = \begin{pmatrix} \frac{\partial g_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial g_1(\mathbf{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial g_m(\mathbf{x})}{\partial x_n} \end{pmatrix}. \tag{2.14}$$

Hessian. All second-order partial derivatives of an at least two times differentiable scalar-valued function $g : \mathbb{R}^n \to \mathbb{R}$ define the *Hessian*

$$\mathcal{H}_g(\mathbf{x}) := \left(\frac{\partial^2 g(\mathbf{x})}{\partial x_j \partial x_i}\right)_{i,j=1,\dots n} = \begin{pmatrix} \frac{\partial^2 g(\mathbf{x})}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 g(\mathbf{x})}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 g(\mathbf{x})}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 g(\mathbf{x})}{\partial x_n \partial x_n} \end{pmatrix}. \tag{2.15}$$

Note that Jacobian and Hessian are directly related for $\mathcal{J}_{\nabla g}(\mathbf{x}) = \mathcal{H}_g(\mathbf{x})$.

Discretization. To solve derivatives in a numerical way we need to discretize them first. They can be approximated by finite difference schemes. Using step sizes $\mathbf{h} \in \mathbb{N}^2$ as defined in Section 2.1 the approximations of first order derivatives of an image $f : \Omega \to \mathbb{R}$ are given by forward differences, e.g.

$$\frac{\partial f}{\partial x} = f_x \approx \frac{f_{i+1,j} - f_{i,j}}{h_x}, \tag{2.16}$$

$$\frac{\partial f}{\partial y} = f_y \approx \frac{f_{i,j+1} - f_{i,j}}{h_y} \tag{2.17}$$

or central differences, e.g.

$$\frac{\partial f}{\partial x} = f_x \approx \frac{f_{i+1,j} - f_{i-1,j}}{2h_x}, \tag{2.18}$$

$$\frac{\partial f}{\partial y} = f_y \approx \frac{f_{i,j+1} - f_{i,j-1}}{2h_y}. \tag{2.19}$$

Second order derivatives can be approximated with central differences, e.g.

$$\frac{\partial^2 f}{\partial x^2} = f_{xx} \approx \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{h_x^2}, \tag{2.20}$$

$$\frac{\partial^2 f}{\partial y^2} = f_{yy} \approx \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{h_y^2} \tag{2.21}$$

or in case of mixed derivatives, e.g.

$$\begin{aligned} \frac{\partial^2 f}{\partial x \partial y} = (f_y)_x &\approx \frac{(f_y)_{i+1,j} - (f_y)_{i-1,j}}{2h_x} \\ &\approx \frac{f_{i+1,j+1} - f_{i+1,j-1} - f_{i-1,j+1} + f_{i-1,j-1}}{4h_x h_y}. \end{aligned} \tag{2.22}$$

2.4 Calculus of Variations

Variational methods compute solutions by minimizing suitable energy functionals. We introduced such an energy functional in Section 1.1 to solve correspondence problems between images. While functions depend on various scalar values, functionals map entire functions to an output. In order to find such minimizing functions we use the theory of *calculus of variations* [30]. It is concerned with determining extrema of functionals and thus states necessary conditions a minimizer needs to satisfy. The conditions are expressed in form of the so-called *Euler-Lagrange equations*, which are about vanishing of the first variation.

The Euler-Lagrange Equation. Before we go into detail with multiple functions and variables, a simple energy functional in 1-D space is considered first. It is given by

$$E\left(u\right) = \int_a^b F\left(x, u\left(x\right), u_x\left(x\right)\right) \mathrm{d}x \tag{2.23}$$

where $u(x) \in \mathcal{C}^1\left[a, b\right]$ as space of all one time continuously differentiable functions defined on $[a, b] \in \mathbb{R}$. Finding u as minimizer of the functional E with the integrand F comes down to solving a partial differential equation

$$0 = F_u - \frac{\mathrm{d}}{\mathrm{d}x} F_{u_x} \tag{2.24}$$

also known as Euler-Lagrange equation. If we have a strictly convex energy functional E, then u is a global minimum which is moreover unique. Additionally, *Neumann boundary conditions* must hold

$$\begin{aligned} F_{u_x}\left(a, u\left(a\right), u_x\left(a\right)\right) &= 0, \\ F_{u_x}\left(b, u\left(b\right), u_x\left(b\right)\right) &= 0 \end{aligned} \tag{2.25}$$

which specify the values of derivatives of u at the boundary of the closed interval $[a, b]$ [30].

Multiple Functions of Multiple Variables. Instead of using only one function depending on one variable, we extend our previous approach to 2D space and looking for two minimizers u and v as it is the usual case for the methods proposed in this thesis. The new energy functional reads

$$E\left(u, v\right) = \int_\Omega F\left(x, y, u\left(x, y\right), u_x\left(x, y\right), u_y\left(x, y\right), v\left(x, y\right), v_x\left(x, y\right), v_y\left(x, y\right)\right) \mathrm{d}x\mathrm{d}y. \tag{2.26}$$

For each minimizer we get an additional Euler-Lagrange equation. Hence, we have to solve two equations

$$0 = F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y}, \tag{2.27}$$

$$0 = F_v - \frac{\partial}{\partial x} F_{v_x} - \frac{\partial}{\partial y} F_{v_y} \tag{2.28}$$

in the image region Ω. With $\mathbf{n}$ as normal vector at the image boundary $\partial\Omega$ the Neumann boundary condition reads

$$0 = \mathbf{n}^\top \begin{pmatrix} F_{u_x} \\ F_{u_y} \end{pmatrix}, \qquad 0 = \mathbf{n}^\top \begin{pmatrix} F_{v_x} \\ F_{v_y} \end{pmatrix}. \tag{2.29}$$

Functionals with Higher Order Derivatives. We cannot only modify the Euler-Lagrange equations with respect to functions or variables, but also with respect to higher order derivatives. This is necessary if the energy functional contains such derivatives. For an energy functional

$$E(u,v) = \int_\Omega F\left(x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yx}, u_{yy}, v, v_x, v_y, v_{xx}, v_{xy}, v_{yx}, v_{yy}\right) \mathrm{d}x\mathrm{d}y \tag{2.30}$$

the Euler-Lagrange equations are given by

$$0 = F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} + \frac{\partial^2}{\partial x^2} F_{u_{xx}} + \frac{\partial^2}{\partial x \partial y} F_{u_{xy}} + \frac{\partial^2}{\partial y \partial x} F_{u_{yx}} + \frac{\partial^2}{\partial y^2} F_{u_{yy}}, \tag{2.31}$$

$$0 = F_v - \frac{\partial}{\partial x} F_{v_x} - \frac{\partial}{\partial y} F_{v_y} + \frac{\partial^2}{\partial x^2} F_{v_{xx}} + \frac{\partial^2}{\partial x \partial y} F_{v_{xy}} + \frac{\partial^2}{\partial y \partial x} F_{v_{yx}} + \frac{\partial^2}{\partial y^2} F_{v_{yy}} \tag{2.32}$$

with Neumann boundary conditions

$$0 = \mathbf{n}^\top \begin{pmatrix} F_{u_x} - \frac{\partial}{\partial x} F_{u_{xx}} - \frac{\partial}{\partial y} F_{u_{xy}} \\ F_{u_y} - \frac{\partial}{\partial x} F_{u_{yx}} - \frac{\partial}{\partial y} F_{u_{yy}} \end{pmatrix}, \qquad 0 = \mathbf{n}^\top \begin{pmatrix} F_{v_x} - \frac{\partial}{\partial x} F_{v_{xx}} - \frac{\partial}{\partial y} F_{v_{xy}} \\ F_{v_y} - \frac{\partial}{\partial x} F_{v_{yx}} - \frac{\partial}{\partial y} F_{v_{yy}} \end{pmatrix},$$

$$0 = \mathbf{n}^\top \begin{pmatrix} F_{u_{xx}} \\ F_{u_{xy}} \end{pmatrix}, \quad 0 = \mathbf{n}^\top \begin{pmatrix} F_{u_{yx}} \\ F_{u_{yy}} \end{pmatrix}, \quad 0 = \mathbf{n}^\top \begin{pmatrix} F_{v_{xx}} \\ F_{v_{xy}} \end{pmatrix}, \quad 0 = \mathbf{n}^\top \begin{pmatrix} F_{v_{yx}} \\ F_{v_{yy}} \end{pmatrix} \tag{2.33}$$

as derived in [30] with $\mathbf{n}$ as normal vector at the image boundary $\partial\Omega$. Note that we omitted the function's variables in Functional 2.30 for convenience.

2.5 Quality Measures

To evaluate the estimated motion fields in a quantitative way some appropriate metrics are necessary. Most of them rely on having correct reference data (*ground truth*) available and computing differences between this data and the estimations of optical flow vectors. The following two methods are frequently used to measure the quality of resulting motion fields:

1. **Spatiotemporal Average Angular Error (AAE):** With this metric that is used in literature often we measure errors as angular deviations from orientation of estimated and correct motion vectors in space-time. The AAE [31] is defined as

$$\mathrm{AAE}\left(\mathbf{w}_c, \mathbf{w}_e\right) = \frac{1}{N \cdot M} \sum_{i=1}^{N} \sum_{j=1}^{M} \arccos\left(\frac{\mathbf{w}_c^\top \mathbf{w}_e}{|\mathbf{w}_c|\,|\mathbf{w}_e|}\right) \tag{2.34}$$

   with $\mathbf{w}_c = (u^c_{i,j}, v^c_{i,j}, 1)^\top$ as ground truth vector and $\mathbf{w}_e = (u^e_{i,j}, v^e_{i,j}, 1)^\top$ as estimated optical flow vector, respectively. N and M denote the image size in both directions.

2. **Average Endpoint Error (AEE):** The AEE is used to measure differences of the endpoints between estimation and ground truth directly. It is given by

$$\mathrm{AEE}\left(\mathbf{w}_c, \mathbf{w}_e\right) = \frac{1}{N \cdot M} \sum_{i=1}^{N} \sum_{j=1}^{M} |\mathbf{w}_c - \mathbf{w}_e| \tag{2.35}$$

   where $\mathbf{w}_c, \mathbf{w}_e, N$ and M are the same as in the AAE case. Since the AAE only considers angular differences between vectors but not their magnitude, estimated motion vectors may point in the correct direction of the ground truth vectors - especially in cases of large displacements (leads to a low AAE) - but still differ in motion length which is not covered by the AAE. Therefore, the AEE is not only employed during our evaluation but also generally preferred.

After we have learned some principles about continuous and discrete images, discussed higher order derivatives and possible approximations, presented basic aspects with respect to minimization of energy functionals, and chose error metrics for evaluation, let us now deal with details of variational optical flow estimation and how to realize it based on a prototypical approach in the following chapter.

3 Optical Flow

Introduced in Section 1.1, we extend the basic Horn and Schunck model [14] in this chapter using various constancy assumptions in the data term and a slight modification of their regularizer. Following these modifications, we describe a suitable minimization strategy to find minimizing functions based on the Euler-Lagrange equations covered in Section 2.4. Afterwards, we discretize and explain how to solve the final equation system. We recall that the basic variational model has the form

$$E(u,v) = \int_\Omega \underbrace{D(u,v)}_{\text{Data term}} + \alpha \underbrace{R(\nabla u, \nabla v)}_{\text{Regularization term}} \mathrm{d}x\mathrm{d}y. \tag{3.1}$$

Gaussian Presmoothing. Before we go into a detailed description of data and smoothness term though, we introduce a preprocessing step, that is frequently performed before doing optical flow computation. It is about convolving the original image sequence f_0 with a spatial Gaussian K_σ of standard deviation σ (in this context also known as *noise scale* [32]) which smoothes the input images. This *presmoothing* step is given by

$$f = K_\sigma * f_0 \tag{3.2}$$

with $*$ as convolution operator and f denotes smoothed images. Presmoothing is done because: (1) It removes high-frequency noise that can negatively influence the estimation. (2) Images are in $\mathcal{C}^\infty$-space, which is an important property for calculation of derivatives. One has to take care of choosing appropriate values for σ, since if σ is too large, crucial image features may be wiped out and hence the results deteriorate. Figure 3.1 visualizes the presmoothing step and the influence of increasing σ.

3.1 Data Term

The data term incorporates information about assumptions that have to be made to find corresponding features and structures between consecutive images as precise as possible. In doing so, the difficulty is to determine such features, which are robust (*invariant*) to certain alterations of a scene, i.e. they should not change if the illumination of the scene changes. In the following we use non-linearized assumptions to permit the estimation of large displacements. Linearization is then deferred to the numerical scheme to handle the nonconvex energy functional.

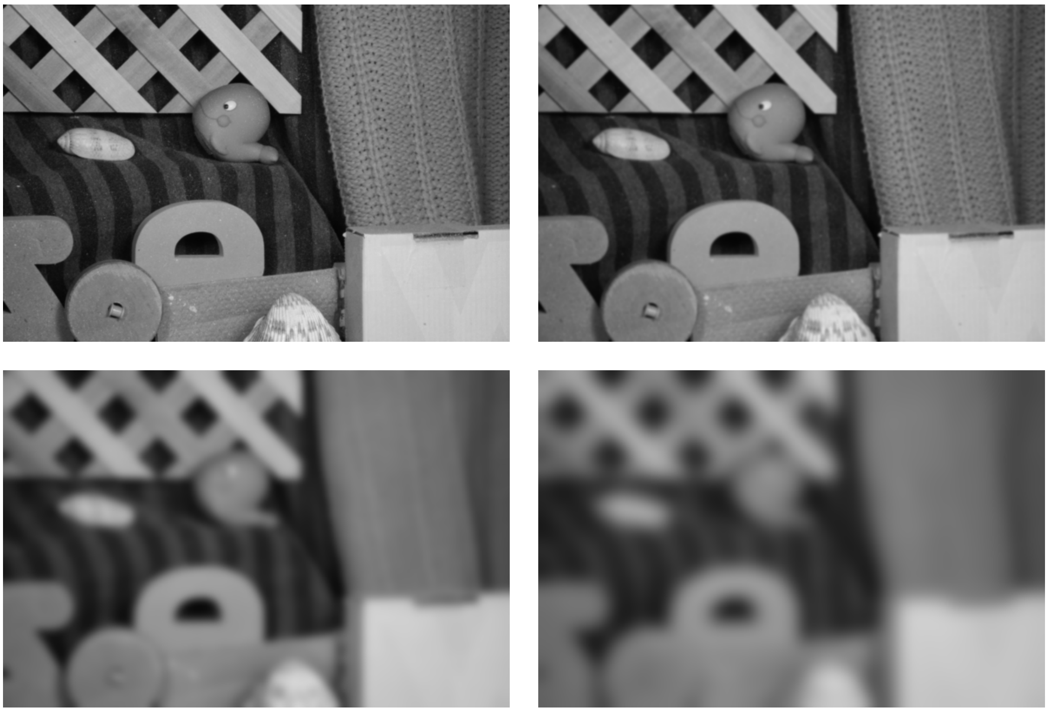

Figure 3.1: Gaussian presmoothing. **Top Left:** Frame 10 of the RubberWhale sequence taken from the Middlebury benchmark data set [33]. **Top Right:** Presmoothed image with $\sigma = 1$. **Bottom Left:** Presmoothed image with $\sigma = 5$. **Bottom Right:** Presmoothed image with $\sigma = 10$.

3.1.1 Gray Value Constancy

Most commonly used in the data term is the gray value constancy assumption. Applying this assumption, the brightness of objects in a scene captured by cameras is supposed to be constant if motion occurs. Thus, our data term reads

$$D_1\left(u,v\right) := \left(f\left(\mathbf{x}+\mathbf{w}\right) - f\left(\mathbf{x}\right)\right)^2 . \tag{3.3}$$

Typically, motion in images can happen for two reasons. The first and intuitive cause is that objects are really moving. Second, due to motion of the capturing camera it only appears as if objects are moving. Since motion - either real or apparent - can cause varying reflections, gray value constancy can be violated in many cases. For example, shadows or shades may arise resulting in slight changes of brightness. Moreover, time differences between captured images can make for various illumination of the scene which also changes gray values. Thus, we need other assumptions that are invariant under gray value changes.

3.1.2 Gradient Constancy

To tackle changing illumination [2] make use of the gradient constancy assumption. Instead of comparing raw gray values, they take their spatial gradients into account now. That results in the new data term

$$D_2\left(u,v\right) := \left|\nabla f\left(\mathbf{x}+\mathbf{w}\right) - \nabla f\left(\mathbf{x}\right)\right|^2. \tag{3.4}$$

This assumption provides invariance under global additive illumination changes, but since gradients contain implicit information about the direction, invariance under rotation is lost. Compared to the gray value constancy it is also more sensitive under noise. Invariance always comes down to losing some kind of information, which makes such descriptors less unique and more ambiguous instead.

3.1.3 Complete Rank Transform

Being robust under global illumination changes while at the same time obtaining maximal amount of local image information, the complete rank transform [18] uses descriptors based on the order of gray values to provide more accurate motion estimation. The basic principle using this constancy assumption is to compare gray values of a neighborhood's pixel, i.e. count all neighbor pixels which have a lower gray value than the reference pixel (*rank transform*). We call this count *rank* of a pixel. To compute the complete rank this transform is applied not only to the reference pixel but also to each pixel in the neighborhood. Eventually, the result is stored in the reference pixel's signature $\mathbf{s}_{\text{CRT}} = \{0,\ldots,\kappa-1\}^\kappa$, where κ denotes the size of a neighborhood. Figure 3.2 shows a visualization of the complete rank transform yielding the signature $\mathbf{s}_{\text{CRT}} = (4,5,13,18,22,2,\ldots,20,0,5,12,17,20)^\top$ for a given pixel. The signature is then used as constancy assumption in the data term

$$D_3\left(u,v\right) := \frac{1}{\kappa}\left|\mathbf{g}\left(\mathbf{x}+\mathbf{w}\right) - \mathbf{g}\left(\mathbf{x}\right)\right|^2 \tag{3.5}$$

with signature length κ and a vector-valued function $\mathbf{g} : \Omega \times [0,\infty) \to \{0,\ldots,\kappa-1\}^\kappa := (g_1,\ldots,g_\kappa)^\top$, that maps the input images to the signatures. The high amount of spatial information in the neighborhood may also cause poor results in regions with rotations due to implicit constraints on feature orientation similar to gradient constancy, but we expect better results for translational and divergent motion because these kinds of motion influence the orientation only slightly.
An overview of the data terms D_1, D_2 and D_3 is given in Table 3.1.

3.1.4 Color Images

So far we discussed data terms with various constancy assumptions based on different image descriptors, but they have one property in common: they all rely on single-channel images (e.g. gray value images) as information source to constrain the optical flow, which limits the estimation accuracy somewhat, since more information can be used if multiple channels

5	15	22	24	37
4	17	18	22	38
4	19	20	25	41
2	19	22	22	33
2	15	21	23	33

4	5	13	18	22
2	7	8	13	23
2	9	11	19	24
0	9	13	13	20
0	5	12	17	20

Figure 3.2: Complete rank transform applied to a gray colored reference pixel with $\kappa = 25$. **Left:** Gray values. **Right:** Complete rank.

are available. Nowadays, even mobile phones have the ability to capture high quality color images. Therefore, we extend our previous data terms to accommodate multi-channel images using the example of gray value constancy. We consider $\mathbf{f}(x, y, t) : \Omega \times [0, \infty) \to \mathbb{R}^n$ as multi-channel images with n different channels $f^1, f^2, \ldots, f^n$. The generalized data term incorporating RGB *color constancy* then reads

$$D_4(u, v) := \sum_{c=1}^{3} \left(f^c(\mathbf{x} + \mathbf{w}) - f^c(\mathbf{x})\right)^2 . \tag{3.6}$$

Gradient constancy and complete rank can be extended analogously:

$$D_5(u, v) := \sum_{c=1}^{3} \left|\nabla f^c(\mathbf{x} + \mathbf{w}) - \nabla f^c(\mathbf{x})\right|^2 , \tag{3.7}$$

$$D_6(u, v) := \frac{1}{\kappa} \sum_{c=1}^{3} \left|\mathbf{g}^c(\mathbf{x} + \mathbf{w}) - \mathbf{g}^c(\mathbf{x})\right|^2 . \tag{3.8}$$

That way the additional constraints of each channel are combined in one data term looking for a motion field that satisfies them jointly.

	Data Term	**Constancy Assumption**	**Illumination invariant**	**Motion Type**
D_1	$(f(\mathbf{x}+\mathbf{w}) - f(\mathbf{x}))^2$	Gray Value	-	any
D_2	$\lvert\nabla f(\mathbf{x}+\mathbf{w}) - \nabla f(\mathbf{x})\rvert^2$	Gradient	global additive	translational divergent
D_3	$\frac{1}{\kappa}\lvert\mathbf{g}(\mathbf{x}+\mathbf{w}) - \mathbf{g}(\mathbf{x})\rvert^2$	Complete Rank	global additive global multiplicative	translational divergent

Table 3.1: Overview of the introduced data terms D_1, D_2 and D_3.

3.1.5 Normalization

By rewriting the quadratic data term, e.g. containing gray value constancy, Zimmer et al. [28] showed that not only the distance between estimated flow $\mathbf{u} := (u,v)^\top$ and normal flow $\mathbf{u}_n$ (see Section 1.1) is penalized as in the ideal case, but the data term is also implicitly weighted by the squared spatial image gradient ∇f:

$$\begin{aligned} D_1(u,v) &= \left(\nabla f^\top \mathbf{u} + f_t\right)^2 \\ &= \left[|\nabla f| \left(\frac{\nabla f^\top \mathbf{u}}{|\nabla f|} + \frac{f_t}{|\nabla f|}\right)\right]^2 \\ &= |\nabla f|^2 \left[\frac{\nabla f^\top}{|\nabla f|}\left(\mathbf{u} + \frac{f_t \nabla f}{|\nabla f|^2}\right)\right]^2 \\ &= \underbrace{|\nabla f|^2}_{\substack{\text{implicit}\\ \text{weight}}} \Bigg(\underbrace{\frac{\nabla f^\top}{|\nabla f|}(\mathbf{u} - \mathbf{u}_n)}_{=:d}\Bigg)^2 . \end{aligned} \tag{3.9}$$

Note that we used a linearized data term to illustrate the problem. The distance d can be interpreted as the projected difference between $\mathbf{u}$ and $\mathbf{u}_n$ in direction of the gradient. Due to this implicit weighting the data term has a varying influence on the solution depending on image regions with different gradients, e.g. larger ones caused by noise. We do not want noise and other unreliable structures to have such an increased effect on the estimation, hence [28] propose a *normalization* by multiplying the data term with a factor

$$\theta_\bullet^c := \frac{1}{|\nabla \star_\bullet^c|^2 + \zeta^2} \tag{3.10}$$

with $\star_\bullet^c \in \left\{f^c, f_x^c, f_y^c, g_k^c\right\}$, $k \in \{1,\dots,\kappa\}$ and a small $\zeta > 0$ to avoid division by 0. Using a more generalized notation the normalized data terms incorporating color information for the various constancy assumptions read

$$D_7(u,v) := \sum_{c=1}^{3} \theta^c \left(f^c(\mathbf{x}+\mathbf{w}) - f^c(\mathbf{x})\right)^2 , \tag{3.11}$$

$$D_8(u,v) := \sum_{c=1}^{3} \sum_{*\in\{x,y\}} \theta_*^c \left(f_*^c(\mathbf{x}+\mathbf{w}) - f_*^c(\mathbf{x})\right)^2 , \tag{3.12}$$

$$D_9(u,v) := \frac{1}{\kappa} \sum_{c=1}^{3} \sum_{k=1}^{\kappa} \theta_k^c \left(g_k^c(\mathbf{x}+\mathbf{w}) - g_k^c(\mathbf{x})\right)^2 . \tag{3.13}$$

The generalized notation reveals that the structure of the data terms is rather similar, which allows straightforward extensions.

3.1.6 Robustification

Until now we penalized deviations from our previous constancy assumptions in a quadratic way. Considering noise and other image distortions or preservation of discontinuities, it is of great interest to reduce the influence of such outliers on the estimation. This can be achieved by using subquadratic penalizing functions $\Psi\left(s^2\right)$, e.g. the regularized L_1 norm

$$\Psi\left(s^2\right) := \sqrt{s^2 + \epsilon^2} \tag{3.14}$$

with a small regularization parameter $\epsilon > 0$ [8] to guarantee a strictly convex and differentiable penalizer after linearization of the respective constancy assumption.
In Figure 3.3 different plots of some penalizers are visualized. In addition to quadratic and linear penalization, an example of a nonconvex function is given. Nonconvex penalizers cause the final energy functional to have multiple local minima and thus complicate to find a globally minimal solution. For that reason such penalizers are usually avoided. Applying Function 3.14 to the quadratic data terms D_7 (color constraints), D_8 (gradient constraints) and D_9 (rank constraints) yield the new linear data terms

$$D_{10}\left(u,v\right) := \Psi\left(\sum_{c=1}^{3} \theta^c \left(f^c\left(\mathbf{x}+\mathbf{w}\right) - f^c\left(\mathbf{x}\right)\right)^2 \right), \tag{3.15}$$

$$D_{11}\left(u,v\right) := \Psi\left(\sum_{c=1}^{3} \sum_{*\in\{x,y\}} \theta^c_* \left(f^c_*\left(\mathbf{x}+\mathbf{w}\right) - f^c_*\left(\mathbf{x}\right)\right)^2 \right), \tag{3.16}$$

$$D_{12}\left(u,v\right) := \Psi\left(\frac{1}{\kappa} \sum_{c=1}^{3} \sum_{k=1}^{\kappa} \theta^c_k \left(g^c_k\left(\mathbf{x}+\mathbf{w}\right) - g^c_k\left(\mathbf{x}\right)\right)^2 \right). \tag{3.17}$$

which are more robust regarding outliers. Since multiple constancy constraints, e.g. color images, are combined and their deviations are penalized together here, we speak of *joint robustification* [2]. Joint robustification makes the most sense if the constraints depend on each other in some way. For example, many advanced techniques for optical flow estimation combine gray value and gradient constancy in the data term [15, 28]. Since these assumptions may hold independently, a *separate robustification* [20] is favored.

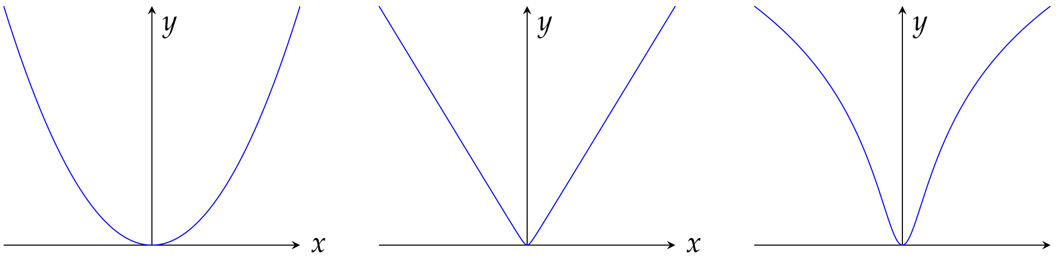

Figure 3.3: Various penalizing functions Ψ in comparison. **Left:** Quadratic (Tikhonov [34]). **Center:** Linear. **Right:** Nonconvex function.

Another suitable subquadratic penalizer which is frequently used is the so-called *Charbonnier* penalizer [35, 36]

$$\begin{aligned}\Psi_{\mathrm{CH}}\left(s^2\right) &:= 2\epsilon^2\sqrt{1+\frac{s^2}{\epsilon^2}}-2\epsilon^2\\ &= 2\epsilon\sqrt{\epsilon^2+s^2}-2\epsilon^2.\end{aligned} \tag{3.18}$$

Rearranging the equation and neglecting constant terms shows that it can be seen as an ϵ-scaled version of the TV penalizer.

An overview of the introduced modifications to the example of gradient constancy is given in Table 3.2.

3.1.7 Quantitative Results

Let us now compare our introduced constancy constraints including the various modifications by estimating the optical flow in some first experiments. To this end, Functional 3.1 serves as our basic framework in which the just presented data terms are incorporated. The smoothness term of the functional realizes homogeneous regularization (see Section 1.1, Equation 1.7 [14]). To visualize the resulting motion field we choose a color representation with color indicating the direction of motion and its brightness denoting the corresponding magnitude of the direction. As evaluation data we use Frame 10 and 11 of the RubberWhale sequence taken from the Middlebury benchmark data set [33]. By choosing this particular sequence the estimation has to face translational and rotational motion under changing illumination. To make the data term robust under outliers we selected the Charbonnier penalizer with $\epsilon = 0.00003$ and for normalization we set $\zeta = 0.01$. Furthermore, the neighborhood created by the complete rank transform has size $\kappa = 9$. Computed optical flows are presented in Figure 3.4 with corresponding parameter values shown in Table 3.3, where we used the average angular error (AAE) and the average endpoint error (AEE) as error metrics to determine differences between estimation and ground truth. Our modifications were applied consecutively, i.e. the last experiment of each constancy assumption consisted of color channels, normalization and penalization applied together to the data term.

Modified Data Term	**Modification**	**Purpose**
$\sum_{c=1}^{3}\sum_{*\in\{x,y\}}\left(f_*^c(\mathbf{x}+\mathbf{w})-f_*^c(\mathbf{x})\right)^2$	Color Images	multiple channels to utilize more information
$\sum_{c=1}^{3}\sum_{*\in\{x,y\}}\theta_*^c\left(f_*^c(\mathbf{x}+\mathbf{w})-f_*^c(\mathbf{x})\right)^2$	Normalization	remove implicit gradient weighting
$\Psi\left(\sum_{c=1}^{3}\sum_{*\in\{x,y\}}\theta_*^c\left(f_*^c(\mathbf{x}+\mathbf{w})-f_*^c(\mathbf{x})\right)^2\right)$	Robustification	reduce influence of outliers

Table 3.2: Overview of the introduced modifications to data terms with gradient constancy.

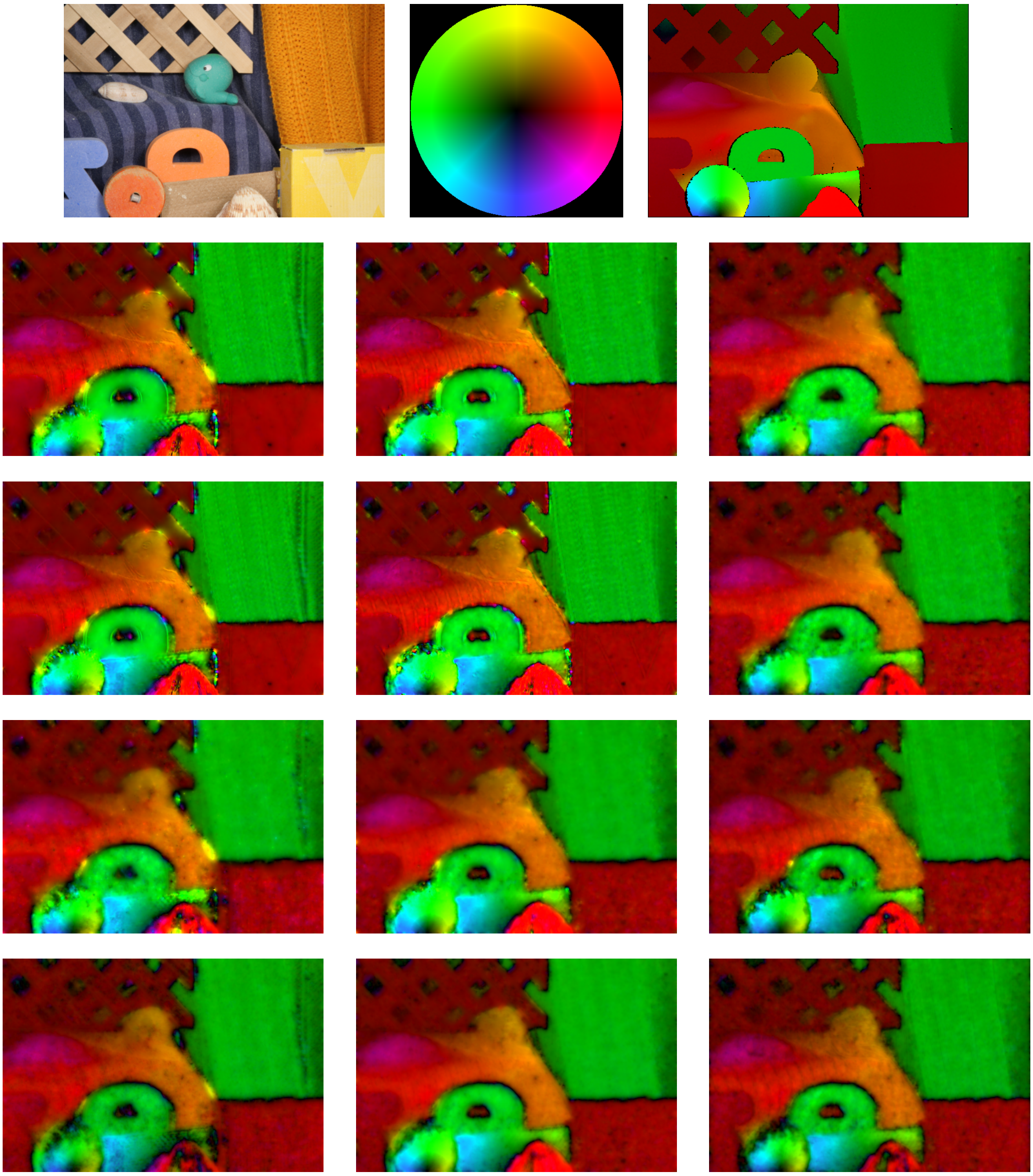

Figure 3.4: Different constancy assumptions with varying data term modifications for translational and rotational motion under changing illumination in comparison. **First Row, Left to Right:** Frame 10 of the RubberWhale sequence with 584 × 388 pixels, color map [8] and ground truth. **Second Row, Left to Right:** Gray value constancy, gradient constancy and complete rank for gray value images. **Third Row, Left to Right:** Gray value constancy, gradient constancy and complete rank applied to color channels. **Fourth Row:** Previous constancy assumptions on color channels modified by normalization. **Last Row:** Normalized constancy assumptions on color channels penalized by Charbonnier.

Constancy Assumption	Modifications	α	σ	AAE	AEE
Gray Value	-	109	0.9	8.9231°	0.3165
	Color Channels	137	0.9	*8.4914°*	*0.3005*
	+ Normalization	40	0.8	9.2441°	0.3230
	+ Robustification	0.00039	0.7	8.9770°	0.3137
Gradient	-	12	1.4	7.7346°	0.2855
	Color Channels	10	1.4	7.4020°	0.2768
	+ Normalization	62	1.5	7.2790°	0.2498
	+ Robustification	0.0005	1.3	***6.9274°***	***0.2473***
Complete Rank	-	0.5	1.2	7.5092°	0.2799
	Color Channels	4	1.2	7.3387°	0.2680
	+ Normalization	210	1.2	7.0665°	*0.2485*
	+ Robustification	0.00075	1	*7.0080°*	0.2571

Table 3.3: Different constancy assumptions with varying data term modifications for translational and rotational motion under changing illumination in comparison. In this example the parameters α and σ were optimized with respect to the AAE.

Observations. Initially we can see that the gradient constancy and the complete rank transform provide better results than the gray value constancy even without any additional data term modifications. Since our frame pair contains illumination changes, assumptions based on ordinary gray values are not sufficient to provide good estimations and hence perform worse than constraints including neighborhood information (Figure 3.5, red rectangles). With additional color information results of all three methods can be improved slightly due to the existence of more information which may be helpful in some cases, e.g. distinguish between objects of different color but similar gray values.
When it comes to normalization the AAEs in Table 3.3 reveal some interesting insights. If applied in case of gray value constancy the error even increases. This may happen for one reason: During normalization the parameter ζ acts as hard boundary specifying up to which magnitude the implicit gradient weight still has an impact. Using gray value constancy the chosen value might be too small since even smallest fluctuations in the image influence the estimation. In fact, experiments showed a better result (AAE = 8.3353°) if ζ was increased to 3.299 and $\alpha = 7.160$. Nonetheless, we will keep ζ fixed in further experiments.
The impact on the estimation with and without normalization of the gray value constraint is shown in Figure 3.5. We can observe that the difference, especially in the yellow rectangle, is slightly less without normalization. In case of the other two remaining constraints eliminating the implicit weighting with fixed ζ proves to be useful. We can also see that robustification improves accuracy of all approaches. As expected furthermore, performance of gradient constancy and complete rank degrades in regions with rotational motion (green rectangles). The varying magnitudes of α are a result of different value ranges caused by the different assumptions and modifications, especially the ϵ-scaled robustification.

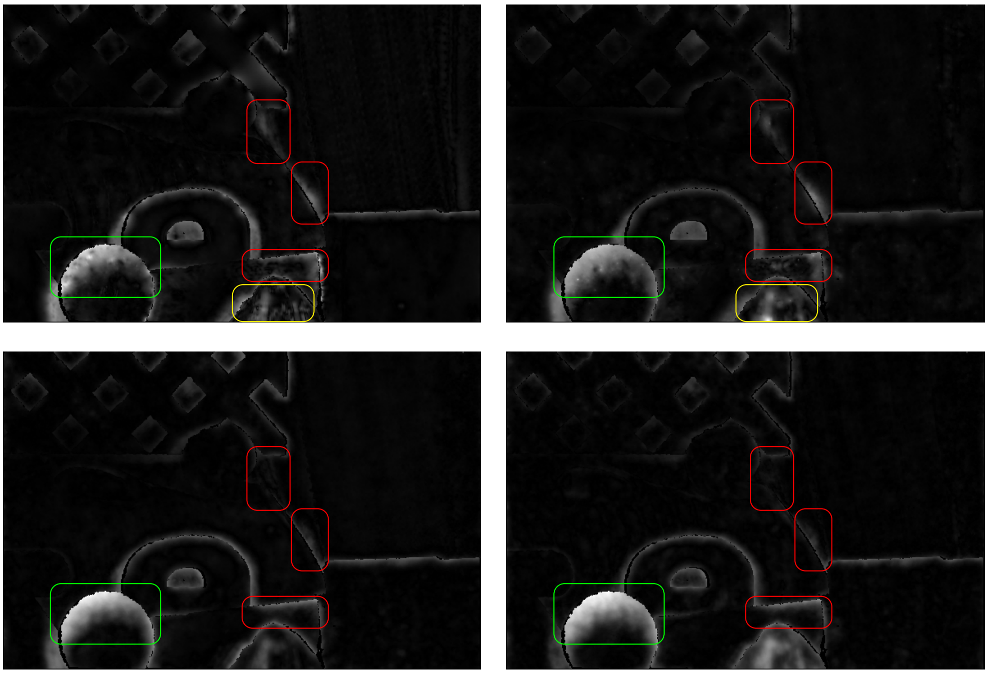

Figure 3.5: Estimation quality of various constancy assumptions visualized as the difference to the ground truth. The brighter the pixel, the higher the difference at this point. **Top Left:** Best result achieved by gray value constancy. **Top Right:** Result achieved by gray value constancy after normalization. **Bottom Left:** Best result achieved by gradient constancy. **Bottom Right:** Best result achieved by complete rank transform.

3.2 Smoothness Term

In the previous section we used homogeneous regularization in form of

$$R_1(\nabla u, \nabla v) := |\nabla u|^2 + |\nabla v|^2 \tag{3.19}$$

to evaluate our proposed data terms. As a reminder, the smoothness term plays an important role in variational motion estimation since it assumes smooth flow fields and helps to overcome the aperture problem by adapting the local solution to the motion of the neighborhood. With homogeneous regularization the flow is treated equally at each point regardless of image structure. Different ideas how to regularize are covered in detail in the upcoming chapters, but nonetheless they share some common properties which are introduced now.

3.2.1 Generic Form and Relation to Diffusion

To come up with a generic form we have to take a first look on the Euler-Lagrange equations presented in Section 2.4. For two minimizing functions u and v they are given by

$$\begin{aligned} 0 &= F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} \\ &= F_u - \operatorname{div} \begin{pmatrix} F_{u_x} \\ F_{u_y} \end{pmatrix}, \qquad (3.20) \\ 0 &= F_v - \frac{\partial}{\partial x} F_{v_x} - \frac{\partial}{\partial y} F_{v_y} \\ &= F_v - \operatorname{div} \begin{pmatrix} F_{v_x} \\ F_{v_y} \end{pmatrix}. \qquad (3.21) \end{aligned}$$

The smoothness term results in a divergence expression, that can be seen as a diffusion process [35, 37]. A generic form of this expression for variational motion estimation reads

$$\operatorname{div}\left(D \cdot \nabla u\right), \qquad (3.22)$$

$$\operatorname{div}\left(D \cdot \nabla v\right). \qquad (3.23)$$

In this context D represents the *diffusion tensor*, which is a symmetric and positive semidefinite 2×2 matrix. It is used to steer the strength and direction of the diffusion process. For homogeneous regularization the diffusion tensor is the identity matrix

$$D_{\mathrm{H}} := I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad (3.24)$$

yielding the divergence

$$\begin{aligned} \operatorname{div}\left(D_{\mathrm{H}} \cdot \nabla u\right) &= \operatorname{div} \begin{pmatrix} u_x \\ u_y \end{pmatrix} \\ &= u_{xx} + u_{yy} \\ &= \triangle u, \qquad (3.25) \\ \operatorname{div}\left(D_{\mathrm{H}} \cdot \nabla v\right) &= \operatorname{div} \begin{pmatrix} v_x \\ v_y \end{pmatrix} \\ &= v_{xx} + v_{yy} \\ &= \triangle v. \qquad (3.26) \end{aligned}$$

3.2.2 Robustification

We already used subquadratic functions $\Psi(s^2)$ in data terms to penalize deviations from model assumptions in a specific way. Similar to constancy constraints such robust functions can be applied to the smoothness term as well to allow flow discontinuities to occur. Preserving such discontinuities is desirable since they usually coincide with object boundaries leading to a more accurate estimation with piecewise smooth flow fields. A modified *isotropic* smoothness term realizing *flow-driven* diffusion reads

$$R_2\left(\nabla u, \nabla v\right) := \Psi\left(|\nabla u|^2 + |\nabla v|^2\right) \tag{3.27}$$

with the diffusion tensor

$$\begin{aligned} D_{1,\mathrm{I}} &:= \Psi'\left(|\nabla u|^2 + |\nabla v|^2\right) \cdot I \\ &= \begin{pmatrix} \Psi'\left(|\nabla u|^2 + |\nabla v|^2\right) & 0 \\ 0 & \Psi'\left(|\nabla u|^2 + |\nabla v|^2\right) \end{pmatrix}. \end{aligned} \tag{3.28}$$

With this modification the diffusion tensor now respects discontinuities in the evolving flow and prevents diffusion at locations where discontinuities (large flow gradients) are present. To do so a suitable function Ψ must be selected. One possible choice for the penalizer Ψ might be the regularized L_1 norm again

$$\Psi_{\mathrm{TV}}\left(s^2\right) := \sqrt{s^2 + \epsilon^2} \tag{3.29}$$

with its derivative

$$\Psi'_{\mathrm{TV}}\left(s^2\right) = \frac{1}{2\sqrt{s^2 + \epsilon^2}} \tag{3.30}$$

inducing the TV diffusivity [21]. But there are more appropriate penalizers available, which we introduce in the following. Also note that due to the penalizers flow-driven regularizers become nonlinear in u and v since they directly influence the corresponding diffusion tensors. We cover that later during minimization.

Charbonnier. A more appropriate penalizer was already mentioned in context of robust data terms (see Subsection 3.1.6). It is the Charbonnier penalizer, a positive and subquadratic function

$$\Psi_{\mathrm{CH}}\left(s^2\right) := 2\epsilon^2\sqrt{1 + \frac{s^2}{\epsilon^2}} - 2\epsilon^2 \tag{3.31}$$

with a positive and decreasing diffusivity function

$$\Psi'_{\mathrm{CH}}\left(s^2\right) = \frac{1}{\sqrt{1 + \frac{s^2}{\epsilon^2}}}. \tag{3.32}$$

Now ϵ serves as a contrast parameter [8]. To examine the behavior of the diffusivity at image edges we can make use of the flux function

$$\Phi_{\mathrm{CH}}\left(s\right) := \Psi'_{\mathrm{CH}}\left(s^2\right) \cdot s \tag{3.33}$$

which equilibrates differences of the flow. We restrict our equation to the 1D case since the behavior also applies to higher dimensional cases [35].

One drawback of the TV diffusivity is its restriction by ϵ for $s^2 \to 0$. That may finally result in very high and varying diffusion at some locations, which degrades estimation quality. Using the Charbonnier penalizer the strength of the diffusion process is always limited to 1 for $s^2 \to 0$ giving a very consistent diffusion. Furthermore, since the flux Φ_{CH} does not have a global maximum, the only behavior at edges is their preservation. Figure 3.6 shows the graphs for Ψ_{CH}, Ψ'_{CH} and Φ_{CH}.

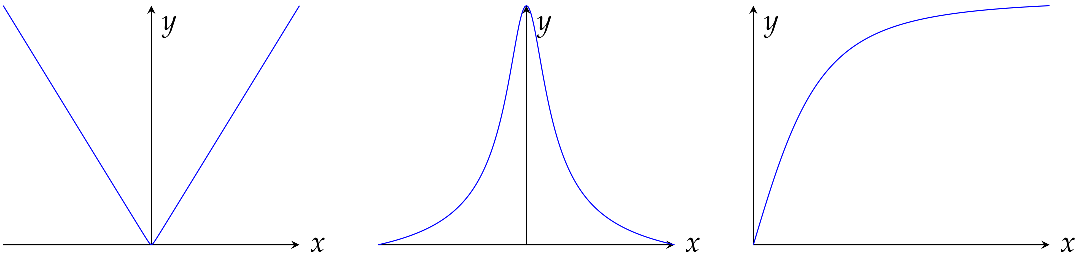

Figure 3.6: Overview of the various Charbonnier functions. **Left:** Charbonnier penalizer Ψ_{CH}. **Center:** Charbonnier diffusivity Ψ'_{CH}. **Right:** Charbonnier flux Φ_{CH}.

Perona-Malik. Another regularizer of choice realizes the so-called *Perona-Malik* diffusivity [38]. The penalizing function reads

$$\Psi_{\mathrm{PM}}\left(s^2\right) := \epsilon^2 \ln\left(1 + \frac{s^2}{\epsilon^2}\right) \tag{3.34}$$

and the diffusivity is given by

$$\Psi'_{\mathrm{PM}}\left(s^2\right) = \frac{1}{1 + \frac{s^2}{\epsilon^2}}. \tag{3.35}$$

The flux is almost the same as before - it only contains Ψ'_{PM} now:

$$\Phi_{\mathrm{PM}}\left(s\right) := \Psi'_{\mathrm{PM}}\left(s^2\right) \cdot s. \tag{3.36}$$

Similar to the Charbonnier penalizer the diffusion process here is once more restricted to strength 1. But the diffusion behavior at image edges differs somewhat compared to Charbonnier. As we can see in Figure 3.7 the flux function now has global maximum determined by the contrast parameter ϵ. In case of $|s| < \epsilon$ we have $\Phi'\left(s\right) > 0$ and thus a *forward diffusion* process. In such regions a usual smoothing behavior occurs. But in regions with $|s| > \epsilon$ a *backward diffusion* behavior sets in since $\Phi'\left(s\right) < 0$. This corresponds to a special smoothing effect that can even enhance edges or other image features [35]. The contrast parameter ϵ balances between these two types of diffusion. Later in this thesis we see application for both the Charbonnier and the Perona-Malik penalizer.

With our data terms and their modifications introduced and the first basic insights into regularizers gained, it remains open how to find the minimizing functions u and v having a concrete discretization. We cover this question in the following sections.

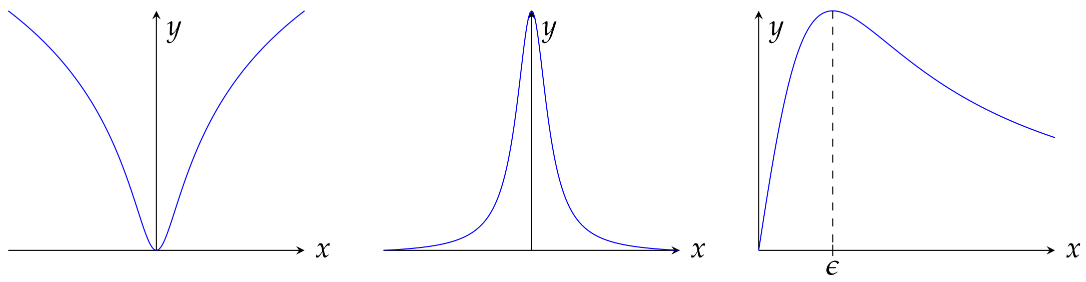

Figure 3.7: Overview of the various Perona-Malik functions. **Left:** Perona-Malik penalizer Ψ_{PM}. **Center:** Perona-Malik diffusivity Ψ'_{PM}. **Right:** Perona-Malik flux Φ_{PM}.

3.3 Minimization

A minimizer of an energy functional must fulfill some necessary conditions as given by the Euler-Lagrange equations introduced in Section 2.4. We recall that for two minimizers u and v and the energy functional

$$E\left(u,v\right) = \int_{\Omega} F\left(x,y,u,u_x,u_y,v,v_x,v_y\right) \mathrm{d}x\mathrm{d}y \tag{3.37}$$

these conditions require the first variations to vanish (see Equation 2.27 and 2.28). Hence, we need the partial derivatives F_u, F_v, F_{u_x}, F_{v_x}, F_{u_y} and F_{v_y}. As we saw in Subsection 3.1.6 the final data terms with all extension are rather similar in structure. The generalized notation allows us to work our way through the minimization using the example of gradient constancy. Euler-Lagrange equations containing color constancy or complete rank constraints then only need some small adjustments. Let us now discuss the partial derivatives with respect to gradient constancy.

Gradient Constancy. Using a *robust* energy functional with *normalized gradient constancy* on *color* channels the partial derivatives yield

$$\begin{aligned} F_u \quad = \quad & 2 \cdot \Psi'\left(\sum_{c=1}^{3} \sum_{*\in\{x,y\}} \theta_*^c \left(f_*^c\left(\mathbf{x}+\mathbf{w}\right) - f_*^c\left(\mathbf{x}\right)\right)^2\right) \\ & \cdot \sum_{c=1}^{3} \sum_{*\in\{x,y\}} \theta_*^c \left(f_*^c\left(\mathbf{x}+\mathbf{w}\right) - f_*^c\left(\mathbf{x}\right)\right) f_{*x}^c\left(\mathbf{x}+\mathbf{w}\right), \\ F_{u_x} = \alpha \cdot 2 \cdot & \Psi'\left(|\nabla u|^2 + |\nabla v|^2\right) \cdot u_x, \\ F_{u_y} = \alpha \cdot 2 \cdot & \Psi'\left(|\nabla u|^2 + |\nabla v|^2\right) \cdot u_y, \end{aligned}$$

$$\begin{aligned}
F_v &= 2 \cdot \Psi'\Bigg(\sum_{c=1}^{3} \sum_{* \in \{x,y\}} \theta_*^c \left(f_*^c\left(\mathbf{x}+\mathbf{w}\right) - f_*^c\left(\mathbf{x}\right)\right)^2\Bigg) \\
&\quad \cdot \sum_{c=1}^{3} \sum_{* \in \{x,y\}} \theta_*^c \left(f_*^c\left(\mathbf{x}+\mathbf{w}\right) - f_*^c\left(\mathbf{x}\right)\right) f_{*y}^c\left(\mathbf{x}+\mathbf{w}\right), \\
F_{v_x} &= \alpha \cdot 2 \cdot \Psi'\Big(|\nabla u|^2 + |\nabla v|^2\Big) \cdot v_x, \\
F_{v_y} &= \alpha \cdot 2 \cdot \Psi'\Big(|\nabla u|^2 + |\nabla v|^2\Big) \cdot v_y. && (3.38)
\end{aligned}$$

Note that we employ $\nabla = \nabla_2$ for simplification. Using the abbreviations of [2, 15]

$$f_z := f\left(\mathbf{x}+\mathbf{w}\right) - f\left(\mathbf{x}\right), \qquad f_{*z} := f_*\left(\mathbf{x}+\mathbf{w}\right) - f_*\left(\mathbf{x}\right), \qquad f_{**} := f_{**}\left(\mathbf{x}+\mathbf{w}\right) \tag{3.39}$$

where $* \in \{x, y\}$ and $** \in \{x, y, xx, xy, yx, yy\}$, the compact Euler-Lagrange equations read

$$\begin{aligned}
0 &= \Psi'\Bigg(\sum_{c=1}^{3} \sum_{* \in \{x,y\}} \theta_*^c \left(f_{*z}^c\right)^2\Bigg) \cdot \sum_{c=1}^{3} \sum_{* \in \{x,y\}} \theta_*^c \left(f_{*z}^c\right) f_{*x}^c \\
&\quad - \alpha \cdot \operatorname{div}\left(D_{1,\mathrm{I}} \cdot \nabla u\right), && (3.40) \\
0 &= \Psi'\Bigg(\sum_{c=1}^{3} \sum_{* \in \{x,y\}} \theta_*^c \left(f_{*z}^c\right)^2\Bigg) \cdot \sum_{c=1}^{3} \sum_{* \in \{x,y\}} \theta_*^c \left(f_{*z}^c\right) f_{*y}^c \\
&\quad - \alpha \cdot \operatorname{div}\left(D_{1,\mathrm{I}} \cdot \nabla v\right) && (3.41)
\end{aligned}$$

with the Neumann boundary conditions

$$0 = \mathbf{n}^\top D_{1,\mathrm{I}} \cdot \nabla u, \qquad\qquad 0 = \mathbf{n}^\top D_{1,\mathrm{I}} \cdot \nabla v. \tag{3.42}$$

The corresponding nonlinear diffusion tensor $D_{1,\mathrm{I}}$ reads the same as in Equation 3.28.

Other Constancy Constraints. Replacing gradient with *color constancy* F_u and F_v change:

$$\begin{aligned}
F_u &= 2 \cdot \Psi'\Bigg(\sum_{c=1}^{3} \theta^c \left(f^c\left(\mathbf{x}+\mathbf{w}\right) - f^c\left(\mathbf{x}\right)\right)^2\Bigg) \cdot \sum_{c=1}^{3} \theta^c \left(f^c\left(\mathbf{x}+\mathbf{w}\right) - f^c\left(\mathbf{x}\right)\right) f_x^c\left(\mathbf{x}+\mathbf{w}\right), \\
F_v &= 2 \cdot \Psi'\Bigg(\sum_{c=1}^{3} \theta^c \left(f^c\left(\mathbf{x}+\mathbf{w}\right) - f^c\left(\mathbf{x}\right)\right)^2\Bigg) \cdot \sum_{c=1}^{3} \theta^c \left(f^c\left(\mathbf{x}+\mathbf{w}\right) - f^c\left(\mathbf{x}\right)\right) f_y^c\left(\mathbf{x}+\mathbf{w}\right). && (3.43)
\end{aligned}$$

Thus, the Euler-Lagrange equations in compact notation for color constancy are given by

$$\begin{aligned}
0 &= \Psi'\Bigg(\sum_{c=1}^{3} \theta^c \left(f_z^c\right)^2\Bigg) \cdot \sum_{c=1}^{3} \theta^c \left(f_z^c\right) f_x^c \\
&\quad - \alpha \cdot \operatorname{div}\left(D_{1,\mathrm{I}} \cdot \nabla u\right), && (3.44) \\
0 &= \Psi'\Bigg(\sum_{c=1}^{3} \theta^c \left(f_z^c\right)^2\Bigg) \cdot \sum_{c=1}^{3} \theta^c \left(f_z^c\right) f_y^c \\
&\quad - \alpha \cdot \operatorname{div}\left(D_{1,\mathrm{I}} \cdot \nabla v\right) && (3.45)
\end{aligned}$$

with the same diffusion tensor and boundary conditions as before. Compared to the previous Euler-Lagrange equations incorporating gradient constancy only the inner sum vanishes.

To perform the *complete rank transform* the Euler-Lagrange equations can be extended similarly. In that case the averaging factor $\frac{1}{\kappa}$ and the rank function $\mathbf{g}$ (to map from the image to the rank signatures) must be considered. Again, only F_u and F_v change:

$$\begin{aligned}
F_u &= 2 \cdot \frac{1}{\kappa} \cdot \Psi' \left(\sum_{c=1}^{3} \sum_{k=1}^{\kappa} \theta_k^c \left(g_k^c \left(\mathbf{x} + \mathbf{w} \right) - g_k^c \left(\mathbf{x} \right) \right)^2 \right) \\
&\quad \cdot \sum_{c=1}^{3} \sum_{k=1}^{\kappa} \theta_k^c \left(g_k^c \left(\mathbf{x} + \mathbf{w} \right) - g_k^c \left(\mathbf{x} \right) \right) g_{kx}^c \left(\mathbf{x} + \mathbf{w} \right), \\
F_v &= 2 \cdot \frac{1}{\kappa} \cdot \Psi' \left(\sum_{c=1}^{3} \sum_{k=1}^{\kappa} \theta_k^c \left(g_k^c \left(\mathbf{x} + \mathbf{w} \right) - g_k^c \left(\mathbf{x} \right) \right)^2 \right) \\
&\quad \cdot \sum_{c=1}^{3} \sum_{k=1}^{\kappa} \theta_k^c \left(g_k^c \left(\mathbf{x} + \mathbf{w} \right) - g_k^c \left(\mathbf{x} \right) \right) g_{ky}^c \left(\mathbf{x} + \mathbf{w} \right).
\end{aligned} \tag{3.46}$$

The abbreviations are extended to include the ranks g_k:

$$g_{kz} := g_k \left(\mathbf{x} + \mathbf{w} \right) - g_k \left(\mathbf{x} \right), \qquad g_{k*} := g_{k*} \left(\mathbf{x} + \mathbf{w} \right) \tag{3.47}$$

where $* \in \{x, y\}$. The compact Euler-Lagrange equations containing complete rank constraints and all of the presented modifications are then given by

$$\begin{aligned}
0 &= \frac{1}{\kappa} \cdot \Psi' \left(\frac{1}{\kappa} \cdot \sum_{c=1}^{3} \sum_{k=1}^{\kappa} \theta_k^c \left(g_{kz}^c \right)^2 \right) \cdot \sum_{c=1}^{3} \sum_{k=1}^{\kappa} \theta_k^c \left(g_{kz}^c \right) g_{kx}^c \\
&\quad - \alpha \cdot \operatorname{div} \left(D_{1,\mathrm{I}} \cdot \nabla u \right),
\end{aligned} \tag{3.48}$$

$$\begin{aligned}
0 &= \frac{1}{\kappa} \cdot \Psi' \left(\frac{1}{\kappa} \cdot \sum_{c=1}^{3} \sum_{k=1}^{\kappa} \theta_k^c \left(g_{kz}^c \right)^2 \right) \cdot \sum_{c=1}^{3} \sum_{k=1}^{\kappa} \theta_k^c \left(g_{kz}^c \right) g_{ky}^c \\
&\quad - \alpha \cdot \operatorname{div} \left(D_{1,\mathrm{I}} \cdot \nabla v \right).
\end{aligned} \tag{3.49}$$

Throughout this chapter we used non-linearized data terms to model large displacements and postponed the linearization up to now. This results in the current energy functional to be *nonconvex*. This property is not desired, since many possible solutions of the Euler-Lagrange equations in terms of multiple local minimizer may exist. Besides, the subquadratic penalizer Ψ is another reason for *nonlinear* data and smoothness terms because of the nonlinear derivative Ψ' no matter what particular penalizer (Total Variation, Charbonnier or Perona-Malik) is chosen. Hence, we need a suitable minimization strategy that covers the nonconvex optimization problem with its nonlinear system of equations. This leads to the incremental coarse-to-fine nested fixed point iteration strategy proposed by Brox et al. [2]. Their strategy comprises three main steps: (1) Fixed point iteration. (2) Incremental computation. (3) Coarse-to-fine strategy. While their strategy is introduced we continue to use gradient constancy in the data term.

3.3.1 Fixed Point Iteration

The first step in the attempt to minimize the nonconvex functional with nonlinear equations is the introduction of a fixed point iteration technique on the argument $\mathbf{w}$, which the image functions implicitly depend on. Let $k \in \mathbb{N}_0$ be the index of the iteration and $\mathbf{w}^k := (u^k, v^k, 1)^\top$ with the initialization $\mathbf{w}^0 := (0, 0, 1)^\top$ at the first iteration step. In addition, $f_\star^k$ are modified abbreviations of Equation 3.39 to contain the index k in their argument $\mathbf{w}^k$. The same applies to the diffusion tensor $D_{1,\mathrm{I}}^k$. To compute $\mathbf{w}^{k+1}$ at iteration k the system of equations

$$\begin{aligned} 0 = \Psi'\left(\sum_{c=1}^{3} \sum_{*\in\{x,y\}} \theta_*^c \left(f_{*z}^{c,k+1}\right)^2\right) \cdot \sum_{c=1}^{3} \sum_{*\in\{x,y\}} \theta_*^c \left(f_{*z}^{c,k+1}\right) f_{*x}^{c,k} \\ - \alpha \cdot \operatorname{div}\left(D_{1,\mathrm{I}}^{k+1} \cdot \nabla u^{k+1}\right), \end{aligned} \tag{3.50}$$

$$\begin{aligned} 0 = \Psi'\left(\sum_{c=1}^{3} \sum_{*\in\{x,y\}} \theta_*^c \left(f_{*z}^{c,k+1}\right)^2\right) \cdot \sum_{c=1}^{3} \sum_{*\in\{x,y\}} \theta_*^c \left(f_{*z}^{c,k+1}\right) f_{*y}^{c,k} \\ - \alpha \cdot \operatorname{div}\left(D_{1,\mathrm{I}}^{k+1} \cdot \nabla v^{k+1}\right) \end{aligned} \tag{3.51}$$

in case of gradient constancy needs to be solved. Note that a semi-implicit scheme (k and $k+1$) in the data term and a fully implicit scheme (only $k+1$) in the smoothness term is employed to achieve faster convergence of a solution and higher stability [2].

3.3.2 Incremental Computation

With the previous introduction of a fixed point iteration we have to solve a system of equations in each iteration that is still nonlinear however. Nevertheless, the iteration strategy allows us to perform an incremental computation to get rid of the nonlinearity caused by f_z^{k+1} and f_{*z}^{k+1}. For this purpose, we split the unknown flows u^{k+1} and v^{k+1} in the already known motion u^k and v^k from the previous iteration and the unknown motion increments du^k and dv^k from the current iteration:

$$u^{k+1} = u^k + du^k, \tag{3.52}$$

$$v^{k+1} = v^k + dv^k. \tag{3.53}$$

Thereupon, we perform a first order Taylor expansion in the data term with respect to du^k and dv^k. Since we assume the motion increments to be small the linearization

$$\begin{aligned} f_*\left(\mathbf{x} + \mathbf{w}^{k+1}\right) \approx \quad & f_*\left(x + u^k, y + v^k, t+1\right) \\ & + f_{*x}\left(x + u^k, y + v^k, t+1\right) du^k + f_{*y}\left(x + u^k, y + v^k, t+1\right) dv^k \\ = \quad & f_*\left(\mathbf{x} + \mathbf{w}^k\right) + f_{*x}\left(\mathbf{x} + \mathbf{w}^k\right) du^k + f_{*y}\left(\mathbf{x} + \mathbf{w}^k\right) dv^k \end{aligned} \tag{3.54}$$

with $f_* \in \{f, f_x, f_y, g_k\}$, $k \in \{1, \ldots, \kappa\}$ is a valid one.

Plugging it into our modified abbreviations yields

$$f_{*z}^{k+1} \approx f_{*z}^{k} + f_{*x}^{k} du^k + f_{*y}^{k} dv^k. \tag{3.55}$$

Equation 3.50 can then be written as

$$\begin{aligned} 0 = \Psi' &\Bigg(\sum_{c=1}^{3} \sum_{* \in \{x,y\}} \theta_*^c \left(f_{*z}^{c,k} + f_{*x}^{c,k} du^k + f_{*y}^{c,k} dv^k \right)^2 \Bigg) \\ &\cdot \sum_{c=1}^{3} \sum_{* \in \{x,y\}} \theta_*^c \left(f_{*z}^{c,k} + f_{*x}^{c,k} du^k + f_{*y}^{c,k} dv^k \right) f_{*x}^{c,k} \\ &- \alpha \cdot \operatorname{div} \left(D_{1,\mathrm{I}}^k \cdot \nabla \left(u^k + du^k \right) \right) \end{aligned} \tag{3.56}$$

with the 2×2 diffusion tensor

$$D_{1,\mathrm{I}}^k := \Psi' \left(\left| \nabla \left(u^k + du^k \right) \right|^2 + \left| \nabla \left(v^k + dv^k \right) \right|^2 \right) \cdot I. \tag{3.57}$$

Equation 3.51 is set up analogously.

To keep our notation even more simple we introduce the motion tensor notation from Bruhn and Weickert [20] at this point. In the following $\mathbf{dw}^k := (du^k, dv^k, 1)^\top$ denotes the motion increment vector and $J_*^k(\mathbf{x} + \mathbf{w}^k)$ describes a symmetric 3×3 motion tensor, which can be seen as a framework for all linearized constancy assumptions. The tensor is defined by

$$J_*^k := \nabla_3 f_* \left(\mathbf{x} + \mathbf{w}^k \right) \nabla_3 f_* \left(\mathbf{x} + \mathbf{w}^k \right)^\top \tag{3.58}$$

with $f_* \in \{f, f_x, f_y, g_k\}$, $k \in \{1, \ldots, \kappa\}$ and the spatiotemporal gradient ∇_3. This tensor is very close to the matrix $f_{*\nabla}^k (f_{*\nabla}^k)^\top$, where $f_{*\nabla}^k := (f_{*x}^k, f_{*y}^k, f_{*z}^k)^\top$, since the difference f_{*z}^k is an approximation of the temporal derivative f_{*t}^k.
Furthermore, we can normalize the motion tensor:

$$\bar{J}_*^k := \theta_* \cdot J_*^k = \theta_* \cdot \nabla_3 f_* \left(\mathbf{x} + \mathbf{w}^k \right) \nabla_3 f_* \left(\mathbf{x} + \mathbf{w}^k \right)^\top. \tag{3.59}$$

By employing the motion tensor notation and setting $\bar{J}_{\mathrm{G}}^k := \sum_{* \in \{x,y\}} \bar{J}_*^k$ as motion tensor for gradient constancy, we are able to rewrite Equation 3.56

$$\begin{aligned} 0 = \Psi' &\left(\left(\mathbf{dw}^k \right)^\top \cdot \sum_{c=1}^{3} \bar{J}_{\mathrm{G}}^{c,k} \cdot \left(\mathbf{dw}^k \right) \right) \cdot \sum_{c=1}^{3} \left(\bar{J}_{\mathrm{G},11}^{c,k} du^k + \bar{J}_{\mathrm{G},12}^{c,k} dv^k + \bar{J}_{\mathrm{G},13}^{c,k} \right) \\ &- \alpha \cdot \operatorname{div} \left(D_{1,\mathrm{I}}^k \cdot \nabla \left(u^k + du^k \right) \right) \end{aligned} \tag{3.60}$$

being a partly linearized fixed point iteration with respect to $\mathbf{dw}^k$ now, which allows to compute a unique solution of a *convex* problem in each step provided that Ψ' is convex.

Due to the derivatives of the penalizer functions Ψ' in the data and smoothness term the newly gained systems of equations remain to be nonlinear in the increments du^k and dv^k, but not in $\mathbf{w}^k$ anymore. Similar as before, a second (inner) fixed point iteration can be employed to overcome the remaining non-linearity. To this end, we introduce a second iteration variable l, that denotes a certain iteration step and enables us to evaluate Ψ' with values from the previous iteration step. Let $\mathbf{dw}^{k,l} := (du^{k,l}, dv^{k,l}, 1)^\top$ with the initialization $\mathbf{dw}^{k,0} := (0,0,1)^\top$ denote the motion increments at a certain iteration l. We also modify the diffusion tensor $D^{k,l}_{1,\mathrm{I}}$ to use $du^{k,l}$ and $dv^{k,l}$.

The finally *linear* system of equations that determines $du^{k,l}$ and $dv^{k,l}$ is then given by

$$\begin{aligned} 0 = \Psi' &\left(\left(\mathbf{dw}^{k,l}\right)^\top \cdot \sum_{c=1}^{3} \bar{J}^{c,k}_{\mathrm{G}} \cdot \left(\mathbf{dw}^{k,l}\right) \right) \cdot \sum_{c=1}^{3} \left(\bar{J}^{c,k}_{\mathrm{G},11} du^{k,l+1} + \bar{J}^{c,k}_{\mathrm{G},12} dv^{k,l+1} + \bar{J}^{c,k}_{\mathrm{G},13} \right) \\ &- \alpha \cdot \operatorname{div}\left(D^{k,l}_{1,\mathrm{I}} \cdot \nabla \left(u^k + du^{k,l+1} \right) \right), \end{aligned} \tag{3.61}$$

$$\begin{aligned} 0 = \Psi' &\left(\left(\mathbf{dw}^{k,l}\right)^\top \cdot \sum_{c=1}^{3} \bar{J}^{c,k}_{\mathrm{G}} \cdot \left(\mathbf{dw}^{k,l}\right) \right) \cdot \sum_{c=1}^{3} \left(\bar{J}^{c,k}_{\mathrm{G},12} du^{k,l+1} + \bar{J}^{c,k}_{\mathrm{G},22} dv^{k,l+1} + \bar{J}^{c,k}_{\mathrm{G},23} \right) \\ &- \alpha \cdot \operatorname{div}\left(D^{k,l}_{1,\mathrm{I}} \cdot \nabla \left(v^k + dv^{k,l+1} \right) \right). \end{aligned} \tag{3.62}$$

If we want to employ color constancy or rank constraints now, only the motion tensors of the system above need to be changed. In case of color constancy the motion tensor is given by $\bar{J}_{\mathrm{C}} := \bar{J}$, whereas $\bar{J}_{\mathrm{R}} := \sum_{k=1}^{\kappa} \bar{J}_k$ incorporates the linearized complete rank constancy assumption.

After each outer iteration step k we have to evaluate $f_*(\mathbf{x} + \mathbf{w}^{k+1})$ in the motion tensor with the new flow again. We can achieve this by compensating the second image of the image pair by the new estimation of the flow. The motion compensation step is also known as *warping* an image. Because the motion increments are computed with subpixel precision, an interpolation such as bilinear interpolation is hereby necessary.

3.3.3 Coarse-To-Fine Strategy

Since the underlying energy functional $E(u,v)$ is nonconvex, the outer fixed point iteration may get trapped in local minima, which are possibly very close to the initialization of the solution. The hierarchical coarse-to-fine strategy with multiple level of different image scales is chosen to alleviate this problem. This approach bases upon building image pyramids with a refinement factor $\eta \in (0,1)$ that specifies how much an image is successively downsampled from a fine to a next coarser level. We then embed our outer fixed point iteration into the image pyramid. By looking at our latest system of equations (Equation 3.61 and 3.62) it becomes evident that image data f is only used in the motion tensor. The modified tensor notation

$$\bar{J}^k_* := \theta^k_* \cdot J^k_* = \theta^k_* \cdot \nabla_3 f^k_* \left(\mathbf{x} + \mathbf{w}^k \right) \nabla_3 f^k_* \left(\mathbf{x} + \mathbf{w}^k \right)^\top \tag{3.63}$$

directly connects a downsampled image f^k_* from hierarchy level k to a corresponding iteration step k, where θ^k also uses different image scales. In other words: Each step k of the outer iteration is performed with the image data taken from the pyramid at level k beginning at the coarsest level $k = 0$. After the inner iteration of a certain outer step has completed the resulting flow is upsampled to serve as initialization of the next finer level.

What is left now is a suitable discretization and the selection of an appropriate method to solve the series of linear systems of equations. These points are covered in the next sections.

3.4 Discretization

To solve the linear equations derived in Section 3.3 numerically, we need to discretize its various components. Therefore, the notation of function values at discretized locations introduced in Section 2.1 is employed. Since we compute the unknown flows u and v as solutions on rectangular images, the minimizing functions can be regarded on rectangular grids, i.e.

$$\begin{aligned} [u]_{i,j} &= u_{i,j} = u\left([(x,y)]_{i,j}\right), \\ [v]_{i,j} &= v_{i,j} = v\left([(x,y)]_{i,j}\right) \end{aligned} \tag{3.64}$$

with the same step sizes $\mathbf{h}^k := (h_x^k, h_y^k)^\top$. Discretizations are necessary of the motion $\mathbf{w}_{i,j}^k$ and its increments $\mathbf{dw}_{i,j}^{k,l}$, the motion tensor (entries) $\bar{J}_{i,j}^k$ with the normalization factor $\theta_{i,j}^k$, the data term penalizer $[(\Psi_D')^{k,l}]_{i,j}$ and the diffusion tensor $[D_{1,1}^{k,l}]_{i,j}$. The therein contained derivatives are thereby approximated using finite difference schemes (see Section 2.3).

Derivatives related to the data term appear in the motion tensor entries

$$\begin{aligned} [\bar{J}_{11}^k]_{i,j} &= [f_x^k]_{i,j} \cdot [f_x^k]_{i,j}, & [\bar{J}_{12}^k]_{i,j} &= [f_x^k]_{i,j} \cdot [f_y^k]_{i,j}, & [\bar{J}_{13}^k]_{i,j} &= [f_x^k]_{i,j} \cdot [f_t^k]_{i,j}, \\ [\bar{J}_{22}^k]_{i,j} &= [f_y^k]_{i,j} \cdot [f_y^k]_{i,j}, & [\bar{J}_{23}^k]_{i,j} &= [f_y^k]_{i,j} \cdot [f_t^k]_{i,j}, & [\bar{J}_{33}^k]_{i,j} &= [f_t^k]_{i,j} \cdot [f_t^k]_{i,j} \end{aligned} \tag{3.65}$$

and the normalization factor (see Equation 3.10). Since the motion tensor is symmetric, we have to compute only six entries effectively. Please note that below the time index t combined with i and j is used to distinguish between the first or the second images of two consecutive images. For derivatives with respect to x and y we use central differences of fourth order

$$[f_x^k]_{i,j} \approx \frac{1}{12h_x^k}\left(-f_{i+2,j,t}^k + 8 \cdot f_{i+1,j,t}^k - 8 \cdot f_{i-1,j,t}^k + f_{i-2,j,t}^k\right), \tag{3.66}$$

$$[f_y^k]_{i,j} \approx \frac{1}{12h_y^k}\left(-f_{i,j+2,t}^k + 8 \cdot f_{i,j+1,t}^k - 8 \cdot f_{i,j-1,t}^k + f_{i,j-2,t}^k\right) \tag{3.67}$$

and forward differences to differentiate with respect to t (h_t is usually set to 1)

$$[f_t^k]_{i,j} \approx \frac{f_{i,j,t+1}^k - f_{i,j,t}^k}{h_t}. \tag{3.68}$$

The remaining derivatives are only contained in the smoothness term. This time we employ nested central differences following [39] to approximate them using the abbreviation

$$\begin{aligned}[(\Psi'_{\mathrm{S}})^{k,l}]_{i,j} := \Psi'_{\mathrm{S}}\Big(&[(u^{k,l} + du^{k,l})_x]^2_{i,j} + [(u^{k,l} + du^{k,l})_y]^2_{i,j} \\ &+ [(v^{k,l} + dv^{k,l})_x]^2_{i,j} + [(v^{k,l} + dv^{k,l})_y]^2_{i,j}\Big)\end{aligned} \tag{3.69}$$

for the penalizer in the diffusion tensor $[D^{k,l}_{1,\mathrm{I}}]_{i,j}$. The derivation is given by

$$\begin{aligned}&-\alpha \cdot \operatorname{div}\left(D^{k,l}_{1,\mathrm{I}} \cdot \nabla\left(u^k + du^{k,l+1}\right)\right) \\ \approx &-\alpha \cdot \sum_{d\in\{x,y\}} \sum_{(\tilde{i},\tilde{j})\in\mathcal{N}_d(i,j)} \frac{[(\Psi'_{\mathrm{S}})^{k,l}]_{\tilde{i},\tilde{j}} + [(\Psi'_{\mathrm{S}})^{k,l}]_{i,j}}{2\cdot\left(h^k_d\right)^2} \cdot \left(u^{k,l}_{\tilde{i},\tilde{j}} + du^{k,l+1}_{\tilde{i},\tilde{j}} - u^{k,l}_{i,j} - du^{k,l+1}_{i,j}\right), &(3.70)\\ &-\alpha \cdot \operatorname{div}\left(D^{k,l}_{1,\mathrm{I}} \cdot \nabla\left(v^k + dv^{k,l+1}\right)\right) \\ \approx &-\alpha \cdot \sum_{d\in\{x,y\}} \sum_{(\tilde{i},\tilde{j})\in\mathcal{N}_d(i,j)} \frac{[(\Psi'_{\mathrm{S}})^{k,l}]_{\tilde{i},\tilde{j}} + [(\Psi'_{\mathrm{S}})^{k,l}]_{i,j}}{2\cdot\left(h^k_d\right)^2} \cdot \left(v^{k,l}_{\tilde{i},\tilde{j}} + dv^{k,l+1}_{\tilde{i},\tilde{j}} - v^{k,l}_{i,j} - dv^{k,l+1}_{i,j}\right) &(3.71)\end{aligned}$$

where $\mathcal{N}_d\,(i,j)$ denotes certain neighbor pixels around a center (i,j) in direction d. Due to the nested approach we have the neighbors $(i+1,j)$ and $(i-1,j)$ along the x-axis and $(i,j+1)$ and $(i,j-1)$ along the y-axis in this particular case.

With all the discrete values at a pixel (i,j) we are by now able to address the problem of solving the discrete system of linear equations

$$\begin{aligned}0 = &[(\Psi'_{\mathrm{D}\star})^{k,l}]_{i,j} \cdot \sum_{c=1}^{3}\left([\bar{J}^{c,k}_{\star,11}]_{i,j} du^{k,l+1}_{i,j} + [\bar{J}^{c,k}_{\star,12}]_{i,j} dv^{k,l+1}_{i,j} + [\bar{J}^{c,k}_{\star,13}]_{i,j}\right) \\ &-\alpha \cdot \sum_{d\in\{x,y\}} \sum_{(\tilde{i},\tilde{j})\in\mathcal{N}_d(i,j)} \frac{[(\Psi'_{\mathrm{S}})^{k,l}]_{\tilde{i},\tilde{j}} + [(\Psi'_{\mathrm{S}})^{k,l}]_{i,j}}{2\cdot\left(h^k_d\right)^2} \cdot \left(u^{k,l}_{\tilde{i},\tilde{j}} + du^{k,l+1}_{\tilde{i},\tilde{j}} - u^{k,l}_{i,j} - du^{k,l+1}_{i,j}\right), &(3.72)\\ 0 = &[(\Psi'_{\mathrm{D}\star})^{k,l}]_{i,j} \cdot \sum_{c=1}^{3}\left([\bar{J}^{c,k}_{\star,12}]_{i,j} du^{k,l+1}_{i,j} + [\bar{J}^{c,k}_{\star,22}]_{i,j} dv^{k,l+1}_{i,j} + [\bar{J}^{c,k}_{\star,23}]_{i,j}\right) \\ &-\alpha \cdot \sum_{d\in\{x,y\}} \sum_{(\tilde{i},\tilde{j})\in\mathcal{N}_d(i,j)} \frac{[(\Psi'_{\mathrm{S}})^{k,l}]_{\tilde{i},\tilde{j}} + [(\Psi'_{\mathrm{S}})^{k,l}]_{i,j}}{2\cdot\left(h^k_d\right)^2} \cdot \left(v^{k,l}_{\tilde{i},\tilde{j}} + dv^{k,l+1}_{\tilde{i},\tilde{j}} - v^{k,l}_{i,j} - dv^{k,l+1}_{i,j}\right) &(3.73)\end{aligned}$$

for the entire image domain, where $\star \in \{\mathrm{C},\mathrm{G},\mathrm{R}\}$ to differentiate between the various constancy assumptions and

$$[(\Psi'_{\mathrm{D}\star})^{k,l}]_{i,j} := \Psi'\left(\left(\mathbf{dw}^{k,l}_{i,j}\right)^{\top} \cdot \sum_{c=1}^{3}[\bar{J}^{c,k}_{\star}]_{i,j} \cdot \left(\mathbf{dw}^{k,l}_{i,j}\right)\right). \tag{3.74}$$

3.5 Solving the System of Equations

The previous linear system of equations 3.72 and 3.73 has the form $A\mathbf{x} = \mathbf{b}$ with $\mathbf{x} := (\mathbf{u}, \mathbf{v})^\top$ having the unknown flow fields stacked on each other and a sparse, yet large matrix A. Since A grows with increasing input image size and today's cameras already provide high resolution images, direct methods, e.g. the Gauss-Elimination method, are not efficient enough to compute the numerical solution of the system. Methods that are more appropriate for such systems rely on iterative computation. Similar as in case of the fixed point iteration strategy in Section 3.3 they successively approximate the solution $\mathbf{x}$. One of such iterative methods is the so-called *Gauss-Seidel method*. It decomposes the matrix A into several parts, where one part should be a close approximation of A and its inverse is used together with the remaining parts during computation. To speed up the relatively slow converging process of the Gauss-Seidel method we use a more sophisticated method that modifies Gauss-Seidel by pointwise extrapolating its result per iteration step m. This technique is the *successive overrelaxation method* (SOR) [40]. A relaxation factor $\omega \in [0, 2)$ determines the strength of the extrapolation. If $\omega = 1$, we have Gauss-Seidel again. Inserting an additional third iteration within the second iteration l the corresponding iteration step for u and v is given by

$$\begin{aligned} du_{i,j}^{k,l+1,m+1} = (1-\omega) \cdot du_{i,j}^{k,l+1,m} + \omega \cdot & \Bigg(-[(\Psi'_{\mathrm{D}\star})^{k,l}]_{i,j} \cdot \sum_{c=1}^{3} \left([\bar{J}^{c,k}_{\star,12}]_{i,j} dv_{i,j}^{k,l+1,m} + [\bar{J}^{c,k}_{\star,13}]_{i,j} \right) \\ & + \alpha \cdot \sum_{d \in \{x,y\}} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_d^-(i,j)} \widetilde{\Psi}^{k,l}_{\tilde{i},\tilde{j},i,j,d} \cdot \left(u^{k,l}_{\tilde{i},\tilde{j}} + du^{k,l+1,m+1}_{\tilde{i},\tilde{j}} - u^{k,l}_{i,j} \right) \\ & + \alpha \cdot \sum_{d \in \{x,y\}} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_d^+(i,j)} \widetilde{\Psi}^{k,l}_{\tilde{i},\tilde{j},i,j,d} \cdot \left(u^{k,l}_{\tilde{i},\tilde{j}} + du^{k,l+1,m}_{\tilde{i},\tilde{j}} - u^{k,l}_{i,j} \right) \Bigg) \\ & \cdot \left([(\Psi'_{\mathrm{D}\star})^{k,l}]_{i,j} \cdot \sum_{c=1}^{3} \left([\bar{J}^{c,k}_{\star,11}]_{i,j} \right) + \alpha \cdot \sum_{d \in \{x,y\}} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_d(i,j)} \widetilde{\Psi}^{k,l}_{\tilde{i},\tilde{j},i,j,d} \right)^{-1}, \end{aligned} \tag{3.75}$$

$$\begin{aligned} dv_{i,j}^{k,l+1,m+1} = (1-\omega) \cdot dv_{i,j}^{k,l+1,m} + \omega \cdot & \Bigg(-[(\Psi'_{\mathrm{D}\star})^{k,l}]_{i,j} \cdot \sum_{c=1}^{3} \left([\bar{J}^{c,k}_{\star,12}]_{i,j} du_{i,j}^{k,l+1,m+1} + [\bar{J}^{c,k}_{\star,23}]_{i,j} \right) \\ & + \alpha \cdot \sum_{d \in \{x,y\}} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_d^-(i,j)} \widetilde{\Psi}^{k,l}_{\tilde{i},\tilde{j},i,j,d} \cdot \left(v^{k,l}_{\tilde{i},\tilde{j}} + dv^{k,l+1,m+1}_{\tilde{i},\tilde{j}} - v^{k,l}_{i,j} \right) \\ & + \alpha \cdot \sum_{d \in \{x,y\}} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_d^+(i,j)} \widetilde{\Psi}^{k,l}_{\tilde{i},\tilde{j},i,j,d} \cdot \left(v^{k,l}_{\tilde{i},\tilde{j}} + dv^{k,l+1,m}_{\tilde{i},\tilde{j}} - v^{k,l}_{i,j} \right) \Bigg) \\ & \cdot \left([(\Psi'_{\mathrm{D}\star})^{k,l}]_{i,j} \cdot \sum_{c=1}^{3} \left([\bar{J}^{c,k}_{\star,22}]_{i,j} \right) + \alpha \cdot \sum_{d \in \{x,y\}} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_d(i,j)} \widetilde{\Psi}^{k,l}_{\tilde{i},\tilde{j},i,j,d} \right)^{-1} \end{aligned} \tag{3.76}$$

with $du_{i,j}^{k,l+1,0} = dv_{i,j}^{k,l+1,0} := 0$ as initialization.

We use the abbreviation

$$\widetilde{\Psi}^{k,l}_{\tilde{i},\tilde{j},i,j,d} := \frac{[(\Psi'_{\mathrm{S}})^{k,l}]_{\tilde{i},\tilde{j}} + [(\Psi'_{\mathrm{S}})^{k,l}]_{i,j}}{2 \cdot \left(h^k_d\right)^2} \tag{3.77}$$

to make the equations more compact. Besides, there are two new sets of neighbor pixels: $\mathcal{N}_d^-(i,j)$ denotes neighbor pixels around a center (i,j) in direction d that were already computed in the current iteration step m and $\mathcal{N}_d^+(i,j)$ denotes neighbor pixels around a center (i,j) in direction d that are yet to be computed in the current iteration step m.
Since optical flow computation and optimization can take a long time during evaluation, we decrease the amount of computation time using a modified variant of the SOR method, where the pixels are traversed in a different order. This allows to perform a more efficient parallelization later. To implement parallel processing we execute multiple threads (instructions blocks) simultaneously on the one hand, and employ vectorization in terms of the SIMD (Single Instruction, Multiple Data) paradigm on the other hand. Using vectorization we can apply the same operation to multiple pixel data at once (data parallelism) instead of calculating values for each pixel separately.

Various aspects of variational optical flow computation have been introduced in this chapter. We discussed different constancy assumptions, several modifications to the data term and their impact on the accuracy of estimation, had a brief introduction to the smoothness term and its relation to diffusion, covered a suitable minimization strategy to handle a nonlinear nonconvex optimization problem and eventually presented how to solve the discrete linear system of equations stemming from the Euler-Lagrange equations. With all this broad new knowledge we will focus on different first and second regularizers and their properties entirely in the next chapter.

4 Regularizers in Detail

In the previous chapter we set the foundation for our taxonomy to evaluate variational optical flow computation with various data terms and regularizers of different order and different concept. In particular, varying constancy assumptions and techniques to estimate the motion fields have already been covered. Hence, we discuss some possible changes to the smoothness term in this chapter to further improve estimation accuracy.

While each of the following models contains a new idea, they all are available in an isotropic and an anisotropic setting. By applying a subquadratic penalizer to the smoothness term as introduced in Subsection 3.2.2 we already showed how to modify a homogeneous regularizer to adapt to flow discontinuities and thus achieve location-dependent smoothing (flow-driven isotropic diffusion). In this case, the diffusion process is equal in all directions, but inhibited at locations with large flow gradients to preserve the discontinuities. However, the resulting flow edges appear to be blurry and therefore not well localized. To obtain sharper flow edges one can steer the diffusion process by incorporating directional information gained from local image structures (*image-driven anisotropic* diffusion), i.e. enforce smoothing along image edges and reduce it across them. Since image-driven approaches suffer from oversegmentation problems in heavily textured areas (image edges do not always coincide with object boundaries, but flow edges usually do), combining both methods leads to sharp flow edges without oversegmentation in the optimal case.

Directional Information. A first attempt of joint image- and flow-driven regularization combining directional information and flow magnitude was performed by Sun et al. [41]. They use the structure tensor [29]

$$J_\rho := K_\rho * \left(\nabla_2 f \nabla_2 f^\top\right) \tag{4.1}$$

to extract the directional information in form of its eigenvectors. The standard deviation ρ and the convolution operator $*$ denote a convolution with a Gaussian K_ρ to integrate directional information of a certain neighborhood. This approach allows to identify image structures such as corners or edges. Since J_ρ is a symmetric positive semidefinite 2×2 matrix with corresponding eigenvalues $\mu_1 \geq \mu_2 \geq 0$, its eigenvectors $\mathbf{e}_1$ and $\mathbf{e}_2$ are orthonormal to each other. The larger eigenvalue μ_1 corresponds to $\mathbf{e}_1$ that points across image edges and $\mathbf{e}_2$ points along them.

The structure tensor in Equation 4.1 only makes use of image brightness making its eigenvector directions inconsistent with more advanced data terms. Thus, Zimmer et al. [28] suggest to use the eigenvectors $\mathbf{r}_1$ and $\mathbf{r}_2$ extracted from a modified variant of the motion tensor instead, that can be considered as a generalized structure tensor. In contrast to the normalized motion tensor $\overline{J}$, the so-called *regularization tensor* R_ρ employs spatial gradients ∇_2 due to spatial regularization and Gaussian convolution to integrate directional information. Setting up n constraints on image features $\star_i$ the regularization tensor in our case is given by

$$R_\rho := \sum_{c=1}^{3} K_\rho * \left(\sum_{i=1}^{n} \theta_i^c \left(\nabla_2 \star_i^c \nabla_2 \star_i^{c\top} \right) \right) . \tag{4.2}$$

For example, the regularization tensor for gradient constancy reads

$$R_{\mathrm{C},\rho} = \sum_{c=1}^{3} K_\rho * \left(\theta_x^c \left(\nabla_2 f_x^c \nabla_2 f_x^{c\top} \right) + \theta_y^c \left(\nabla_2 f_y^c \nabla_2 f_y^{c\top} \right) \right) . \tag{4.3}$$

Since we are only interested in directional information from the first frame, only this frame is employed to compute the central differences as discretization of the derivatives in the regularization tensor.

4.1 First Order

The regularizer that has been applied during minimization in Chapter 3 contained first order derivatives to penalize jumps in the estimated solution, thus allowing it to model piecewise smooth flow fields. This is enforced by favoring motion fields with spatial variations close to zero in a certain region, i.e. the motion is constant within this area. If objects move parallel to the capturing camera (fronto-parallel motion), the property of constant motion holds and hence we may expect good results in such scenarios. For the sake of completeness we list the isotropic variant once more.

4.1.1 Isotropic

The well-known isotropic regularizer that penalizes spatial gradients of u and v reads

$$R_{\mathrm{I,I}}(u,v) := \int_\Omega \Psi \left(|\nabla u|^2 + |\nabla v|^2 \right) \mathrm{d}x\mathrm{d}y \tag{4.4}$$

where $\Psi := \Psi_{\mathrm{CH}}$. Discretization aspects are omitted here, since they were already dealt with before. Rudin et al. [21] proposed to penalize the spatial variations in such a way in context of noise removal first. But instead of Ψ_{CH} they used the total variation penalizer Ψ_{TV}. The corresponding stencil based on the isotropic diffusion tensor

$$D_{1,\mathrm{I}} := \begin{pmatrix} \Psi'\left(|\nabla u|^2 + |\nabla v|^2\right) & 0 \\ 0 & \Psi'\left(|\nabla u|^2 + |\nabla v|^2\right) \end{pmatrix} \tag{4.5}$$

is shown in Table A.1 in Appendix A.

4.1.2 Anisotropic

An anisotropic extension to the isotropic regularizer is the complementary regularizer

$$R_{\mathrm{I,A}}\left(u,v\right) := \int_{\Omega} \Psi_1\left(\left(\mathbf{r}_1^\top \nabla u\right)^2 + \left(\mathbf{r}_1^\top \nabla v\right)^2\right) + \Psi_2\left(\left(\mathbf{r}_2^\top \nabla u\right)^2 + \left(\mathbf{r}_2^\top \nabla v\right)^2\right) \mathrm{d}x\mathrm{d}y \tag{4.6}$$

presented by Zimmer et al. [15] with joint penalization of spatial derivatives of u and v in the eigenvector directions. We set $\Psi_1 := \Psi_{\mathrm{PM}}$ to exploit the Perona-Malik backward diffusion behavior to enhance edges. Along flow edges we penalize with $\Psi_2 := \Psi_{\mathrm{CH}}$ again. The new anisotropic diffusion tensor is then given by

$$D_{1,\mathrm{A}} := (\mathbf{r}_1|\mathbf{r}_2)\begin{pmatrix} \Psi_1'((\mathbf{r}_1^\top\nabla u)^2 + (\mathbf{r}_1^\top\nabla v)^2) & 0 \\ 0 & \Psi_2'((\mathbf{r}_2^\top\nabla u)^2 + (\mathbf{r}_2^\top\nabla v)^2) \end{pmatrix}\begin{pmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \end{pmatrix}. \tag{4.7}$$

If we denote the motion tensor entries as

$$D_{1,\mathrm{A}} := \begin{pmatrix} a & b \\ b & c \end{pmatrix} \tag{4.8}$$

and write out the divergence expression 3.22 (3.23 is written out analogously)

$$\begin{aligned} \operatorname{div}\left(D_{1,\mathrm{A}} \cdot \nabla u\right) = \operatorname{div} &\begin{pmatrix} au_x + bu_y \\ bu_x + cu_y \end{pmatrix} \\ &= \partial_x(au_x) + \partial_x(bu_y) + \partial_y(bu_x) + \partial_y(cu_y), \end{aligned} \tag{4.9}$$

it becomes evident that - in contrast to the isotropic case - the entry $b \neq 0$ and thus the mixed partial derivatives $\partial_x(bu_y)$ and $\partial_y(bu_x)$ remain. An approach with usual discretization may result in a negative discretization of $\operatorname{div}(D_{1,\mathrm{A}} \cdot \nabla u)$. Therefore, we employ a stencil (see Table A.2) to discretize the divergence expressions in a particular way that was introduced in [42].
To have a first impression on the motion estimation of both first order isotropic and anisotropic regularizers, we show the resulting flow fields in Figure 4.1.

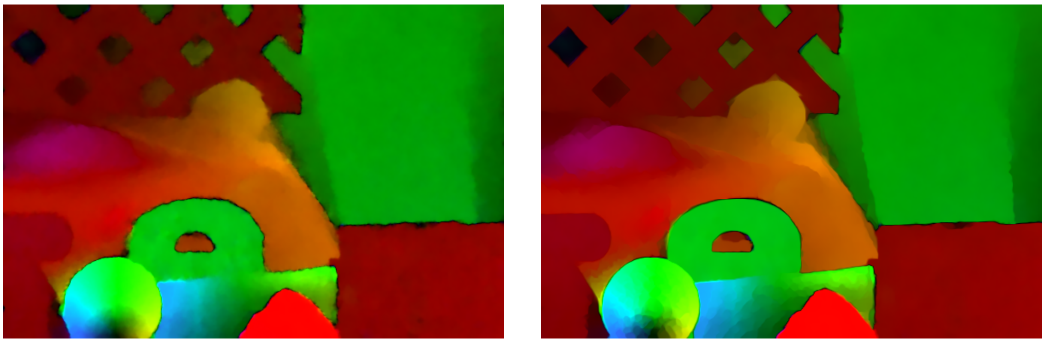

Figure 4.1: Comparison of the first order regularizers. **Left:** Isotropic. **Right:** Anisotropic.

4.2 Second Order

While first order derivatives are suitable to model fronto-parallel motion, they lack the ability to consider motion that changes in an *affine* way. Such type of motion may be caused e.g. by the egomotion of the capturing camera. Hence, we present regularizers based on second order derivatives in this section. If we enforce the second order derivatives to be close to zero (low variation), first order derivatives in a certain region are allowed to reach similar values different from zero. This corresponds to kinks in the estimated flow field and thus instead of implicitly preferring constant flow fields, piecewise affine solutions are favored. Same as in Section 4.1 we consider second order isotropic and anisotropic regularizers.

4.2.1 Isotropic

In contrast to Trobin et al. [24], we measure deviations from second derivatives using the Frobenius norm of the Hessian of u and v in the same way as Demetz et al. [23] already did before. The second order isotropic regularizer reads

$$R_{\mathrm{II,I}}\left(u,v\right) := \int_{\Omega} \Psi\left(\|\mathcal{H}_u\|_{\mathrm{F}}^2 + \|\mathcal{H}_v\|_{\mathrm{F}}^2\right) \mathrm{d}x\mathrm{d}y. \tag{4.10}$$

Once more we penalize with $\Psi := \Psi_{\mathrm{CH}}$. The second order derivatives change the energy functional and the Euler-Lagrange equations as described in Equation 2.31 and 2.32 to

$$0 = F_{\mu} - \frac{\partial}{\partial x}F_{\mu_x} - \frac{\partial}{\partial y}F_{\mu_y} + \frac{\partial^2}{\partial x^2}F_{\mu_{xx}} + \frac{\partial^2}{\partial x \partial y}F_{\mu_{xy}} + \frac{\partial^2}{\partial y \partial x}F_{\mu_{yx}} + \frac{\partial^2}{\partial y^2}F_{\mu_{yy}} \tag{4.11}$$

with the new derivatives of the integrand

$$\begin{aligned}
F_{\mu_x} &= 0,\\
F_{\mu_{xx}} &= \alpha \cdot 2 \cdot \Psi'\left(\|\mathcal{H}_u\|_{\mathrm{F}}^2 + \|\mathcal{H}_v\|_{\mathrm{F}}^2\right) \cdot \mu_{xx},\\
F_{\mu_{xy}} &= \alpha \cdot 2 \cdot \Psi'\left(\|\mathcal{H}_u\|_{\mathrm{F}}^2 + \|\mathcal{H}_v\|_{\mathrm{F}}^2\right) \cdot \mu_{xy},\\
F_{\mu_y} &= 0,\\
F_{\mu_{yx}} &= \alpha \cdot 2 \cdot \Psi'\left(\|\mathcal{H}_u\|_{\mathrm{F}}^2 + \|\mathcal{H}_v\|_{\mathrm{F}}^2\right) \cdot \mu_{yx},\\
F_{\mu_{yy}} &= \alpha \cdot 2 \cdot \Psi'\left(\|\mathcal{H}_u\|_{\mathrm{F}}^2 + \|\mathcal{H}_v\|_{\mathrm{F}}^2\right) \cdot \mu_{yy}.
\end{aligned} \tag{4.12}$$

Please note that we use $\mu \in \{u, v\}$ here and in the following because partial derivatives of u and v can be set analogously. Since we do not penalize first order derivatives anymore, F_{μ_x} and F_{μ_y} become zero. That way our divergence expression also changes

$$\mathrm{div}_2\left(D_{2,\mathrm{I}} \cdot \begin{pmatrix}\nabla \mu_x \\ \nabla \mu_y\end{pmatrix}\right) \tag{4.13}$$

where $\mathrm{div}_2 := (\partial_{xx}, \partial_{xy}, \partial_{yx}, \partial_{yy})$ and

$$D_{2,\mathrm{I}} := \begin{pmatrix} \Psi' & 0 & 0 & 0 \\ 0 & \Psi' & 0 & 0 \\ 0 & 0 & \Psi' & 0 \\ 0 & 0 & 0 & \Psi' \end{pmatrix} \tag{4.14}$$

with $\Psi' := \Psi'(\|\mathcal{H}_u\|_\mathrm{F}^2 + \|\mathcal{H}_v\|_\mathrm{F}^2)$. The Neumann boundary conditions are given by

$$0 = \mathbf{n}^\top(\Psi' \cdot \nabla\mu_x), \qquad 0 = \mathbf{n}^\top(\Psi' \cdot \nabla\mu_y), \qquad 0 = \mathbf{n}^\top \begin{pmatrix} \mathrm{div}(\Psi' \cdot \nabla\mu_x) \\ \mathrm{div}(\Psi' \cdot \nabla\mu_y) \end{pmatrix}. \tag{4.15}$$

Table A.3 shows the stencil for the corresponding discretization of the new second order divergence expression with its derivation presented in Section B.1.

4.2.2 Anisotropic

The second order isotropic smoothness term can be extended to perform a direction-dependent regularization in a similar fashion as the first order isotropic one. We incorporate the directional information in terms of the eigenvectors as follows:

$$R_{\mathrm{II,A}}(u,v) := \int_\Omega \Psi_1\left(|\mathcal{H}_u\mathbf{r}_1|^2 + |\mathcal{H}_v\mathbf{r}_1|^2\right) + \Psi_2\left(|\mathcal{H}_u\mathbf{r}_2|^2 + |\mathcal{H}_v\mathbf{r}_2|^2\right) \mathrm{d}x\mathrm{d}y. \tag{4.16}$$

As before we set $\Psi_1 := \Psi_{\mathrm{PM}}$ and $\Psi_2 := \Psi_{\mathrm{CH}}$. The new diffusion tensor inducing anisotropic behavior is given by

$$D_{2,\mathrm{A}} := \begin{pmatrix} r_{11} & r_{21} & 0 & 0 \\ r_{12} & r_{22} & 0 & 0 \\ 0 & 0 & r_{11} & r_{21} \\ 0 & 0 & r_{12} & r_{22} \end{pmatrix} \begin{pmatrix} \Psi_1' & 0 & 0 & 0 \\ 0 & \Psi_2' & 0 & 0 \\ 0 & 0 & \Psi_1' & 0 \\ 0 & 0 & 0 & \Psi_2' \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & 0 & 0 \\ r_{21} & r_{22} & 0 & 0 \\ 0 & 0 & r_{11} & r_{12} \\ 0 & 0 & r_{21} & r_{22} \end{pmatrix} \tag{4.17}$$

where $\Psi_k' := \Psi_k'(|\mathcal{H}_u\mathbf{r}_k|^2 + |\mathcal{H}_v\mathbf{r}_k|^2)$ with $k \in \{1,2\}$. Compared to the isotropic case, the Neumann boundary conditions contain the first 2×2 part of $D_{2,\mathrm{A}}$ that is defined by

$$D_{2,\mathrm{A}}^2 := (\mathbf{r}_1|\mathbf{r}_2) \begin{pmatrix} \Psi_1'(|\mathcal{H}_u\mathbf{r}_1|^2 + |\mathcal{H}_v\mathbf{r}_1|^2) & 0 \\ 0 & \Psi_2'(|\mathcal{H}_u\mathbf{r}_2|^2 + |\mathcal{H}_v\mathbf{r}_2|^2) \end{pmatrix} \begin{pmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \end{pmatrix}. \tag{4.18}$$

Stencil and derivation of the associated standard discretization are shown in Table A.4 and Section B.2, respectively.
Again, we have a first look on the results of the motion estimation using the second order isotropic or anisotropic regularizer in Figure 4.2. The differences between both approaches are more evident as in the first order case with this data set.

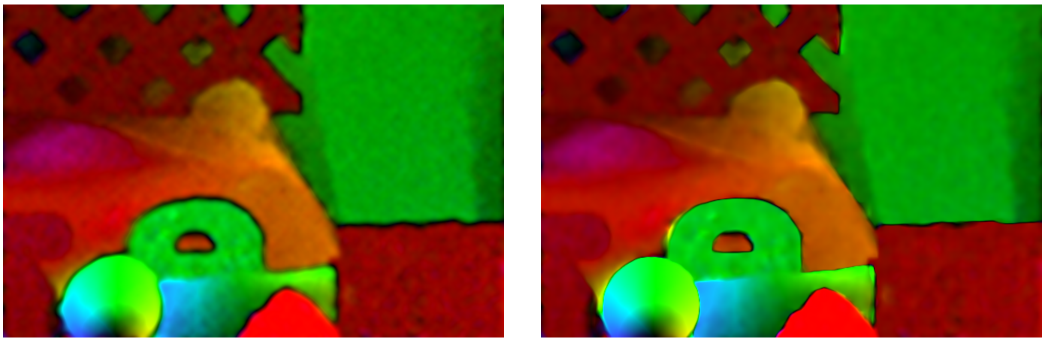

Figure 4.2: Comparison of the second order regularizers. **Left:** Isotropic. **Right:** Anisotropic.

4.3 First and Second Order Combined

First order regularizers allow to model motion edges at objects boundaries as we saw in Section 4.1. They also, however, tend to produce artifacts within an object, which is referred to as the staircasing phenomenon (Figure 4.3, red rectangle). In contrast, second order regularization is used to consider affine changes of the motion field. Since it favors such affine solutions, artifacts appear in regions with usually constant motion (Figure 4.3, green rectangle). While first order regularizers can produce constant flow fields with sharp edges, second order ones do not suffer from staircasing effects. Thus, to overcome the drawbacks of both approaches we combine penalization of first and second order derivatives in an isotropic and anisotropic *infimal convolution* approach. This prior was first used by Chambolle and Lions [22] in the field of image denoising to reduce occurring artifacts.

4.3.1 Isotropic

With infimal convolution we can no longer refer to the regularization term and the smoothness term as being equivalent. The regularization term now as a general container comprises all aspects related to constraining the final form of the flow field. The smoothness term still measures deviations from assumptions on the field.
Hence, our new regularizer consists of two smoothness terms simply combined via addition:

$$R_{\text{III,I}}\left(u,v\right) := \min_{\substack{u_1+u_2=u\\ v_1+v_2=v}} \int_\Omega \Psi\left(|\nabla u_1|^2 + |\nabla v_1|^2\right) + \beta\Psi\left(\|\mathcal{H}_{u_2}\|_{\text{F}}^2 + \|\mathcal{H}_{v_2}\|_{\text{F}}^2\right) \mathrm{d}x\mathrm{d}y \tag{4.19}$$

where $\Psi := \Psi_{\text{CH}}$. We can see that each unknown part of the solution u and v is divided into two flow components u_1 and u_2 or v_1 and v_2, respectively. The first smoothness term with the two flow components u_1 and v_1 realizes a first order penalization (models constant fields), while the second one favors affine fields by penalizing second order derivatives in u_2 and v_2 directly. The flow components of each flow are then added to give the final motion estimation. In addition, the weight β is used to steer the influence of both constraints.

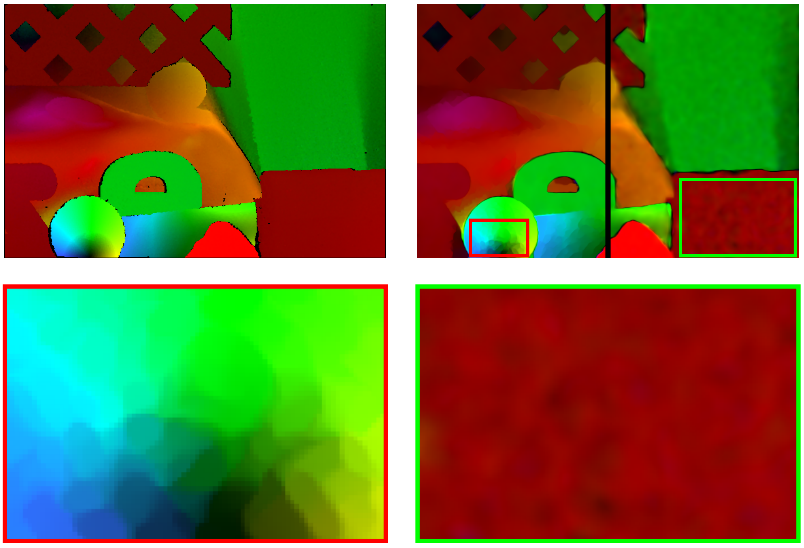

Figure 4.3: Undesirable effects of first and second order regularization. **Top Left:** Ground truth image. **Top Right:** Resulting flows merged to one image separated by the black bar in the middle. **Bottom Left:** Magnification of the staircasing effect (first order). **Bottom Right:** Magnification of the varying affine flow (second order).

Since this regularizer now contains derivatives of both orders and four flow components, we have the partial derivatives (only the relevant ones are listed)

$$\begin{aligned}
F_{(\mu_1)_x} &= \alpha \cdot 2 \cdot \Psi' \left(|\nabla u_1|^2 + |\nabla v_1|^2 \right) \cdot (\mu_1)_x \,, \\
F_{(\mu_1)_y} &= \alpha \cdot 2 \cdot \Psi' \left(|\nabla u_1|^2 + |\nabla v_1|^2 \right) \cdot (\mu_1)_y \,, \\
F_{(\mu_2)_{xx}} &= \beta \cdot 2 \cdot \Psi' \left(\|\mathcal{H}_{u_2}\|_F^2 + \|\mathcal{H}_{v_2}\|_F^2 \right) \cdot (\mu_2)_{xx} \,, \\
F_{(\mu_2)_{xy}} &= \beta \cdot 2 \cdot \Psi' \left(\|\mathcal{H}_{u_2}\|_F^2 + \|\mathcal{H}_{v_2}\|_F^2 \right) \cdot (\mu_2)_{xy} \,, \\
F_{(\mu_2)_{yx}} &= \beta \cdot 2 \cdot \Psi' \left(\|\mathcal{H}_{u_2}\|_F^2 + \|\mathcal{H}_{v_2}\|_F^2 \right) \cdot (\mu_2)_{yx} \,, \\
F_{(\mu_2)_{yy}} &= \beta \cdot 2 \cdot \Psi' \left(\|\mathcal{H}_{u_2}\|_F^2 + \|\mathcal{H}_{v_2}\|_F^2 \right) \cdot (\mu_2)_{yy} \qquad (4.20)
\end{aligned}$$

and hence four different Euler-Lagrange equations to solve:

$$0 = F_{u_1} - \alpha \cdot \operatorname{div}\left(D_{1,\mathrm{I}} \cdot \nabla u_1 \right), \qquad (4.21)$$

$$0 = F_{v_1} - \alpha \cdot \operatorname{div}\left(D_{1,\mathrm{I}} \cdot \nabla v_1 \right), \qquad (4.22)$$

$$0 = F_{u_2} + \beta \cdot \operatorname{div}_2 \left(D_{2,\mathrm{I}} \cdot \begin{pmatrix} \nabla (u_2)_x \\ \nabla (u_2)_y \end{pmatrix} \right), \qquad (4.23)$$

$$0 = F_{v_2} + \beta \cdot \operatorname{div}_2 \left(D_{2,\mathrm{I}} \cdot \begin{pmatrix} \nabla (v_2)_x \\ \nabla (v_2)_y \end{pmatrix} \right) \qquad (4.24)$$

where $D_{1,\mathrm{I}}$ contains u_1 and v_1 and $D_{2,\mathrm{I}}$ includes u_2 and v_2, respectively.

The Neumann boundary conditions read

$$\begin{aligned} &0 = \mathbf{n}^\top D_{1,\mathrm{I}} \cdot \nabla \mu_1, \\ &0 = \mathbf{n}^\top (\Psi' \cdot \nabla (\mu_2)_x), \qquad 0 = \mathbf{n}^\top (\Psi' \cdot \nabla (\mu_2)_y), \qquad 0 = \mathbf{n}^\top \begin{pmatrix} \operatorname{div}(\Psi' \cdot \nabla (\mu_2)_x) \\ \operatorname{div}(\Psi' \cdot \nabla (\mu_2)_y) \end{pmatrix} \end{aligned} \tag{4.25}$$

where $\Psi' := \Psi'(\|\mathcal{H}_{u_2}\|_{\mathrm{F}}^2 + \|\mathcal{H}_{v_2}\|_{\mathrm{F}}^2)$. The infimal convolution approach still penalizes first and second order derivatives separately and therefore we can use the same discretization schemes as shown in the previous sections (4.1 and 4.2) for the isotropic cases.

4.3.2 Anisotropic

The anisotropic extension for infimal convolution is given analogously by the regularizer

$$\begin{aligned} R_{\mathrm{III,A}}(u, v) := \min_{\substack{u_1+u_2=u \\ v_1+v_2=v}} \int_\Omega & \Psi_1\Big(\left(\mathbf{r}_1^\top \nabla u_1\right)^2 + \left(\mathbf{r}_1^\top \nabla v_1\right)^2\Big) + \Psi_2\Big(\left(\mathbf{r}_2^\top \nabla u_1\right)^2 + \left(\mathbf{r}_2^\top \nabla v_1\right)^2\Big) \\ & + \beta \Psi_1\Big(|\mathcal{H}_{u_2}\mathbf{r}_1|^2 + |\mathcal{H}_{v_2}\mathbf{r}_1|^2\Big) + \beta \Psi_2\Big(|\mathcal{H}_{u_2}\mathbf{r}_2|^2 + |\mathcal{H}_{v_2}\mathbf{r}_2|^2\Big) \mathrm{d}x\mathrm{d}y \end{aligned} \tag{4.26}$$

with the partial derivatives

$$\begin{aligned} F_{(\mu_1)_x} &= \quad \alpha \cdot 2 \cdot \Psi_1' \left(\left(\mathbf{r}_1^\top \nabla u_1\right)^2 + \left(\mathbf{r}_1^\top \nabla v_1\right)^2\right) \cdot \left(\mathbf{r}_1^\top \nabla \mu_1\right) \cdot r_{11} \\ &\quad + \alpha \cdot 2 \cdot \Psi_2' \left(\left(\mathbf{r}_2^\top \nabla u_1\right)^2 + \left(\mathbf{r}_2^\top \nabla v_1\right)^2\right) \cdot \left(\mathbf{r}_2^\top \nabla \mu_1\right) \cdot r_{21}, \\ F_{(\mu_1)_y} &= \quad \alpha \cdot 2 \cdot \Psi_1' \left(\left(\mathbf{r}_1^\top \nabla u_1\right)^2 + \left(\mathbf{r}_1^\top \nabla v_1\right)^2\right) \cdot \left(\mathbf{r}_1^\top \nabla \mu_1\right) \cdot r_{12} \\ &\quad + \alpha \cdot 2 \cdot \Psi_2' \left(\left(\mathbf{r}_2^\top \nabla u_1\right)^2 + \left(\mathbf{r}_2^\top \nabla v_1\right)^2\right) \cdot \left(\mathbf{r}_2^\top \nabla \mu_1\right) \cdot r_{22}, \\ F_{(\mu_2)_{xx}} &= \quad \beta \cdot 2 \cdot \Psi_1' \left(|\mathcal{H}_{u_2}\mathbf{r}_1|^2 + |\mathcal{H}_{v_2}\mathbf{r}_1|^2\right) \cdot \left(\mathbf{r}_1^\top \nabla (\mu_2)_x\right) \cdot r_{11} \\ &\quad + \beta \cdot 2 \cdot \Psi_2' \left(|\mathcal{H}_{u_2}\mathbf{r}_2|^2 + |\mathcal{H}_{v_2}\mathbf{r}_2|^2\right) \cdot \left(\mathbf{r}_2^\top \nabla (\mu_2)_x\right) \cdot r_{21}, \\ F_{(\mu_2)_{xy}} &= \quad \beta \cdot 2 \cdot \Psi_1' \left(|\mathcal{H}_{u_2}\mathbf{r}_1|^2 + |\mathcal{H}_{v_2}\mathbf{r}_1|^2\right) \cdot \left(\mathbf{r}_1^\top \nabla (\mu_2)_x\right) \cdot r_{12} \\ &\quad + \beta \cdot 2 \cdot \Psi_2' \left(|\mathcal{H}_{u_2}\mathbf{r}_2|^2 + |\mathcal{H}_{v_2}\mathbf{r}_2|^2\right) \cdot \left(\mathbf{r}_2^\top \nabla (\mu_2)_x\right) \cdot r_{22}, \\ F_{(\mu_2)_{yx}} &= \quad \beta \cdot 2 \cdot \Psi_1' \left(|\mathcal{H}_{u_2}\mathbf{r}_1|^2 + |\mathcal{H}_{v_2}\mathbf{r}_1|^2\right) \cdot \left(\mathbf{r}_1^\top \nabla (\mu_2)_y\right) \cdot r_{11} \\ &\quad + \beta \cdot 2 \cdot \Psi_2' \left(|\mathcal{H}_{u_2}\mathbf{r}_2|^2 + |\mathcal{H}_{v_2}\mathbf{r}_2|^2\right) \cdot \left(\mathbf{r}_2^\top \nabla (\mu_2)_y\right) \cdot r_{21}, \\ F_{(\mu_2)_{yy}} &= \quad \beta \cdot 2 \cdot \Psi_1' \left(|\mathcal{H}_{u_2}\mathbf{r}_1|^2 + |\mathcal{H}_{v_2}\mathbf{r}_1|^2\right) \cdot \left(\mathbf{r}_1^\top \nabla (\mu_2)_y\right) \cdot r_{12} \\ &\quad + \beta \cdot 2 \cdot \Psi_2' \left(|\mathcal{H}_{u_2}\mathbf{r}_2|^2 + |\mathcal{H}_{v_2}\mathbf{r}_2|^2\right) \cdot \left(\mathbf{r}_2^\top \nabla (\mu_2)_y\right) \cdot r_{22}. \end{aligned} \tag{4.27}$$

Using the same notation as before (here $D_{1,\mathrm{A}}$ incorporates its corresponding flow components u_1 and v_1, whereas $D_{2,\mathrm{A}}$ includes u_2 and v_2) and $\Psi_1 := \Psi_{\mathrm{PM}}$ as well as $\Psi_2 := \Psi_{\mathrm{CH}}$, the abbreviated Euler-Lagrange equations yield

$$0 = F_{u_1} - \alpha \cdot \operatorname{div}\left(D_{1,\mathrm{A}} \cdot \nabla u_1\right), \tag{4.28}$$

$$0 = F_{v_1} - \alpha \cdot \operatorname{div}\left(D_{1,\mathrm{A}} \cdot \nabla v_1\right), \tag{4.29}$$

$$0 = F_{u_2} + \beta \cdot \operatorname{div}_2\left(D_{2,\mathrm{A}} \cdot \begin{pmatrix} \nabla(u_2)_x \\ \nabla(u_2)_y \end{pmatrix}\right), \tag{4.30}$$

$$0 = F_{v_2} + \beta \cdot \operatorname{div}_2\left(D_{2,\mathrm{A}} \cdot \begin{pmatrix} \nabla(v_2)_x \\ \nabla(v_2)_y \end{pmatrix}\right). \tag{4.31}$$

If we also define

$$D_{2,\mathrm{A}}^2 := (\mathbf{r}_1|\mathbf{r}_2) \begin{pmatrix} \Psi_1'(|\mathcal{H}_{u_2}\mathbf{r}_1|^2 + |\mathcal{H}_{v_2}\mathbf{r}_1|^2) & 0 \\ 0 & \Psi_2'(|\mathcal{H}_{u_2}\mathbf{r}_2|^2 + |\mathcal{H}_{v_2}\mathbf{r}_2|^2) \end{pmatrix} \begin{pmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \end{pmatrix} \tag{4.32}$$

to contain u_2 and v_2 now, the Neumann boundary conditions for anisotropic infimal convolution read

$$\begin{aligned} &0 = \mathbf{n}^\top D_{1,\mathrm{A}} \cdot \nabla \mu_1, \\ &0 = \mathbf{n}^\top D_{2,\mathrm{A}}^2 \cdot \nabla(\mu_2)_x, \quad 0 = \mathbf{n}^\top D_{2,\mathrm{A}}^2 \cdot \nabla(\mu_2)_y, \quad 0 = \mathbf{n}^\top \begin{pmatrix} \operatorname{div}(D_{2,\mathrm{A}}^2 \cdot \nabla(\mu_2)_x) \\ \operatorname{div}(D_{2,\mathrm{A}}^2 \cdot \nabla(\mu_2)_y) \end{pmatrix}. \end{aligned} \tag{4.33}$$

Again, the discretization stencils of the previous anisotropic cases are also valid here. Figure 4.4 shows that regular motion edges get sharpened in the isotropic case, but affine flow fluctuations in usually constant areas remain nearly the same as with second order isotropic regularization. When it comes to the anisotropic regularizer, motion fields estimated with second order regularization and infimal convolution are almost indistinguishable from each other. Only subtle differences exist. Nonetheless, staircasing effects cannot be observed. Please note that this is only the result for one data set so far.

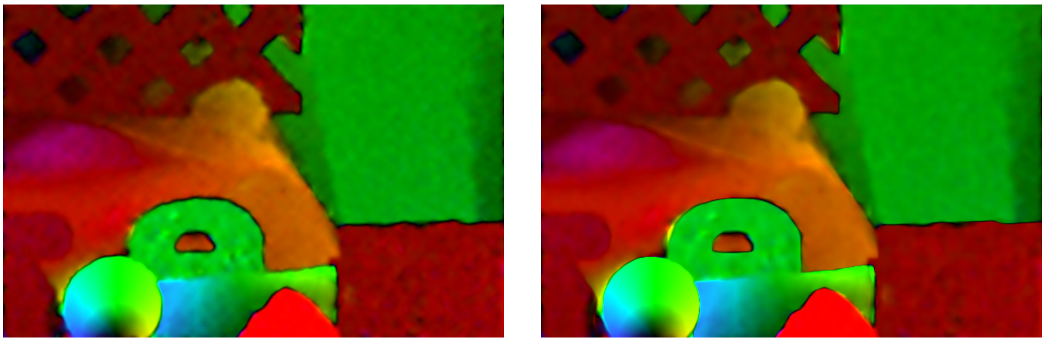

Figure 4.4: Comparison of the regularizers realizing infimal convolution. **Left:** Isotropic. **Right:** Anisotropic.

4.4 Coupled Regularization

Another approach to prevent appearing artifacts is rather similar to the infimal convolution technique but follows another concept to combine first and second order derivatives. Instead of regularizing various flows with respect to derivatives of different order, we employ the usual flow field (u, v) known from the direct regularization approaches in Section 4.1 and 4.2 and couple their gradient fields with some general auxiliary vector fields $\mathbf{p}$ and $\mathbf{q}$. Usually, such a *coupling model* can incorporate variations of higher order, but we restrict ourselves to the case of order 2. That way we are able to perform a direct comparison to the previous infimal convolution method. Again, we differentiate between isotropic and anisotropic variants.
To improve image reconstruction methods like denoising, Bredies et al. [25] introduced their *total generalized variation* based regularization as one of the first techniques performing a coupled regularization.

4.4.1 Isotropic

Compared to infimal convolution the isotropic regularization term now consist of a coupling term that couples the gradient fields to the auxiliary vector fields and a smoothness term that regularizes these vector fields:

$$R_{\mathrm{IV,I}}(u, v) := \min_{\mathbf{p},\mathbf{q}} \int_\Omega \underbrace{\Psi\left(|\nabla u - \mathbf{p}|^2 + |\nabla v - \mathbf{q}|^2\right)}_{\text{Coupling term}} + \beta \underbrace{\Psi\left(\|\mathcal{J}_\mathbf{p}\|_\mathrm{F}^2 + \|\mathcal{J}_\mathbf{q}\|_\mathrm{F}^2\right)}_{\text{Smoothness term}} \mathrm{d}x\mathrm{d}y \qquad (4.34)$$

where $\Psi := \Psi_{\mathrm{CH}}$. Through the coupling term we can see that the vector fields $\mathbf{p} := (p_1, p_2)^\top$ and $\mathbf{q} := (q_1, q_2)^\top$ are indirectly assumed to be approximations of the spatial gradients of u and v. Moreover, second order derivatives are not explicitly formulated but implicitly given by the Jacobian $\mathcal{J}_\mathbf{g}$ for a vector field $\mathbf{g}$. We recall that $\mathcal{J}_{\nabla g} = \mathcal{H}_g$. This may be an advantage during discretization since it is not necessary to derive certain stencils to approximate second order derivatives.
Let $\boldsymbol{\varphi} \in \{\mathbf{p}, \mathbf{q}\}$ in the following. The smoothness term constraints the vector fields $\mathbf{p}$ and $\mathbf{q}$ to have a low variation and thus models affine changes of the final flow field. The coupling term is then responsible to carry this information to u and v by enforcing a low difference between gradient field $\nabla\mu$ and corresponding vector field $\boldsymbol{\varphi}$, but still allowing discontinuities (jumps) at object boundaries to occur by applying a penalizer function. Additionally, it makes the vector field adapt to the better solution of u and v in constant areas, which also fits to the affine change constraint (areas with constant motion also cause second order derivatives to be zero).
We previously defined infimal convolution as

$$\min_{\substack{u_1+u_2=u\\ v_1+v_2=v}} \int_\Omega \Psi\left(|\nabla u_1|^2 + |\nabla v_1|^2\right) + \beta\Psi\left(\|\mathcal{H}_{u_2}\|_\mathrm{F}^2 + \|\mathcal{H}_{v_2}\|_\mathrm{F}^2\right) \mathrm{d}x\mathrm{d}y. \qquad (4.35)$$

If we rewrite the component equations as $u_1 = u - u_2$ and $v_1 = v - v_2$ and use them to substitute Equation 4.35 with, the new definition yields

$$\min_{u_2,v_2} \int_{\Omega} \Psi \left(|\nabla u - \nabla u_2|^2 + |\nabla v - \nabla v_2|^2 \right) + \beta \Psi \left(\|\mathcal{H}_{u_2}\|_{\mathrm{F}}^2 + \|\mathcal{H}_{v_2}\|_{\mathrm{F}}^2 \right) \mathrm{d}x\mathrm{d}y. \tag{4.36}$$

The relation between infimal convolution and coupled regularization becomes evident now. While the spatial gradient in the coupling term is only applied to u and v, the infimal convolution model applies the gradient to u, v, u_2 and v_2. If we assume $\mathbf{p} = \nabla u_2$ and $\mathbf{q} = \nabla v_2$ for a moment, this eventually leads to $\mathcal{J}_{\mathbf{p}} = \mathcal{H}_{u_2}$ and $\mathcal{J}_{\mathbf{q}} = \mathcal{H}_{v_2}$. In this case, coupled regularization reduces to combined regularization. But since $\mathbf{p}$ and $\mathbf{q}$ are general vector fields, whereas the gradient operator enforces u_2 and v_2 to be a specific kind of vector fields (gradient fields), they are not subject to certain limitations of gradient fields. In that sense, general vector fields are more flexible than specific gradient fields since they offer a higher degree of freedom, which might lead to better solutions in the end.
Previously, the first partial derivatives F_u and F_v depended on the data term. Due to the coupling term the first partial derivatives $F_{p_1}, F_{p_2}, F_{q_1}$ and F_{q_2} are now relevant for the regularization as well:

$$\begin{aligned}
F_{u_x} &= \alpha \cdot 2 \cdot \Psi' \left(|\nabla u - \mathbf{p}|^2 + |\nabla v - \mathbf{q}|^2 \right) \cdot (u_x - p_1)\,, \\
F_{u_y} &= \alpha \cdot 2 \cdot \Psi' \left(|\nabla u - \mathbf{p}|^2 + |\nabla v - \mathbf{q}|^2 \right) \cdot (u_y - p_2)\,, \\
F_{v_x} &= \alpha \cdot 2 \cdot \Psi' \left(|\nabla u - \mathbf{p}|^2 + |\nabla v - \mathbf{q}|^2 \right) \cdot (v_x - q_1)\,, \\
F_{v_y} &= \alpha \cdot 2 \cdot \Psi' \left(|\nabla u - \mathbf{p}|^2 + |\nabla v - \mathbf{q}|^2 \right) \cdot (v_y - q_2)\,, \\
F_{p_1} &= \alpha \cdot 2 \cdot \Psi' \left(|\nabla u - \mathbf{p}|^2 + |\nabla v - \mathbf{q}|^2 \right) \cdot (u_x - p_1) \cdot (-1)\,, \\
F_{p_2} &= \alpha \cdot 2 \cdot \Psi' \left(|\nabla u - \mathbf{p}|^2 + |\nabla v - \mathbf{q}|^2 \right) \cdot (u_y - p_2) \cdot (-1)\,, \\
F_{q_1} &= \alpha \cdot 2 \cdot \Psi' \left(|\nabla u - \mathbf{p}|^2 + |\nabla v - \mathbf{q}|^2 \right) \cdot (v_x - q_1) \cdot (-1)\,, \\
F_{q_2} &= \alpha \cdot 2 \cdot \Psi' \left(|\nabla u - \mathbf{p}|^2 + |\nabla v - \mathbf{q}|^2 \right) \cdot (v_y - q_2) \cdot (-1)\,, \\
F_{(\varphi_*)_x} &= \beta \cdot 2 \cdot \Psi' \left(\|\mathcal{J}_{\mathbf{p}}\|_{\mathrm{F}}^2 + \|\mathcal{J}_{\mathbf{q}}\|_{\mathrm{F}}^2 \right) \cdot (\varphi_*)_x, \\
F_{(\varphi_*)_y} &= \beta \cdot 2 \cdot \Psi' \left(\|\mathcal{J}_{\mathbf{p}}\|_{\mathrm{F}}^2 + \|\mathcal{J}_{\mathbf{q}}\|_{\mathrm{F}}^2 \right) \cdot (\varphi_*)_y
\end{aligned} \tag{4.37}$$

where $\varphi_* \in \{p_1, p_2, q_1, q_2\}$. Since we have six fields (u, v, p_1, p_2, q_1 and q_2) in total, we also have six Euler-Lagrange equations to determine (Equation 4.40 and 4.41 contain two equations each):

$$0 = F_u - \alpha \cdot \mathrm{div} \left(D_{\mathrm{C,I}} \cdot (\nabla u - \mathbf{p}) \right), \tag{4.38}$$

$$0 = F_v - \alpha \cdot \mathrm{div} \left(D_{\mathrm{C,I}} \cdot (\nabla v - \mathbf{q}) \right), \tag{4.39}$$

$$0 = \alpha \cdot D_{\mathrm{C,I}} \cdot (\mathbf{p} - \nabla u) - \beta \cdot \mathbf{div} \left(\mathcal{J}_{\mathbf{p}} T_{\mathrm{C,I}} \right), \tag{4.40}$$

$$0 = \alpha \cdot D_{\mathrm{C,I}} \cdot (\mathbf{q} - \nabla v) - \beta \cdot \mathbf{div} \left(\mathcal{J}_{\mathbf{q}} T_{\mathrm{C,I}} \right). \tag{4.41}$$

The diffusion tensors are given by

$$D_{\mathrm{C,I}} := \begin{pmatrix} \Psi'(|\nabla u - \mathbf{p}|^2 + |\nabla v - \mathbf{q}|^2) & 0 \\ 0 & \Psi'(|\nabla u - \mathbf{p}|^2 + |\nabla v - \mathbf{q}|^2) \end{pmatrix} \tag{4.42}$$

and

$$T_{\mathrm{C,I}} := \begin{pmatrix} \Psi'(\|\mathcal{J}_{\mathbf{p}}\|_{\mathrm{F}}^2 + \|\mathcal{J}_{\mathbf{q}}\|_{\mathrm{F}}^2) & 0 \\ 0 & \Psi'(\|\mathcal{J}_{\mathbf{p}}\|_{\mathrm{F}}^2 + \|\mathcal{J}_{\mathbf{q}}\|_{\mathrm{F}}^2) \end{pmatrix}. \tag{4.43}$$

Furthermore, **div** is defined as the row-wise application of the divergence operator div:

$$\mathbf{div}\left(\mathcal{J}_{\mathbf{p}} T\right) = \begin{pmatrix} \operatorname{div}(T \cdot \nabla p_1) \\ \operatorname{div}(T \cdot \nabla p_2) \end{pmatrix}, \qquad \mathbf{div}\left(\mathcal{J}_{\mathbf{q}} T\right) = \begin{pmatrix} \operatorname{div}(T \cdot \nabla q_1) \\ \operatorname{div}(T \cdot \nabla q_2) \end{pmatrix}. \tag{4.44}$$

The Neumann boundary conditions read

$$0 = \mathbf{n}^\top D_{\mathrm{C,I}} \cdot (\nabla u - \mathbf{p}), \qquad 0 = \mathbf{n}^\top D_{\mathrm{C,I}} \cdot (\nabla v - \mathbf{q}), \qquad 0 = \mathbf{n}^\top T_{\mathrm{C,I}} \cdot \nabla \varphi_*. \tag{4.45}$$

To discretize the divergence expressions in Equation 4.38 and 4.39 suitable stencils are given in Table A.5 with their corresponding derivation presented in Section B.3. The weights for the remaining divergence expressions can be found in Stencil A.1.

4.4.2 Anisotropic

Hafner et al. [26] proposed anisotropic diffusion to a coupled regularizer in the field of focus fusion by adding the eigenvectors of the motion tensor. Similarly we incorporate directional information as with our preceding regularizing concepts:

$$\begin{aligned} R_{\mathrm{IV,A}}\left(u,v\right) := \min_{\mathbf{p},\mathbf{q}} \int_\Omega & \Psi_1\left(\left(\mathbf{r}_1^\top\left(\nabla u - \mathbf{p}\right)\right)^2 + \left(\mathbf{r}_1^\top\left(\nabla v - \mathbf{q}\right)\right)^2\right) \\ + \;& \Psi_2\left(\left(\mathbf{r}_2^\top\left(\nabla u - \mathbf{p}\right)\right)^2 + \left(\mathbf{r}_2^\top\left(\nabla v - \mathbf{q}\right)\right)^2\right) \\ & + \beta\Psi_1\left(\left|\mathcal{J}_{\mathbf{p}}\mathbf{r}_1\right|^2 + \left|\mathcal{J}_{\mathbf{q}}\mathbf{r}_1\right|^2\right) + \beta\Psi_2\left(\left|\mathcal{J}_{\mathbf{p}}\mathbf{r}_2\right|^2 + \left|\mathcal{J}_{\mathbf{q}}\mathbf{r}_2\right|^2\right) \mathrm{d}x\mathrm{d}y. \end{aligned} \tag{4.46}$$

As usually we set $\Psi_1 := \Psi_{\mathrm{PM}}$ and $\Psi_2 := \Psi_{\mathrm{CH}}$. The partial derivatives of the regularizer read

$$\begin{aligned} F_{u_x} = \;& \alpha \cdot 2 \cdot \Psi_1'\left(\left(\mathbf{r}_1^\top\left(\nabla u - \mathbf{p}\right)\right)^2 + \left(\mathbf{r}_1^\top\left(\nabla v - \mathbf{q}\right)\right)^2\right) \cdot \left(\mathbf{r}_1^\top\left(\nabla u - \mathbf{p}\right)\right) \cdot r_{11} \\ & + \alpha \cdot 2 \cdot \Psi_2'\left(\left(\mathbf{r}_2^\top\left(\nabla u - \mathbf{p}\right)\right)^2 + \left(\mathbf{r}_2^\top\left(\nabla v - \mathbf{q}\right)\right)^2\right) \cdot \left(\mathbf{r}_2^\top\left(\nabla u - \mathbf{p}\right)\right) \cdot r_{21}, \\ F_{u_y} = \;& \alpha \cdot 2 \cdot \Psi_1'\left(\left(\mathbf{r}_1^\top\left(\nabla u - \mathbf{p}\right)\right)^2 + \left(\mathbf{r}_1^\top\left(\nabla v - \mathbf{q}\right)\right)^2\right) \cdot \left(\mathbf{r}_1^\top\left(\nabla u - \mathbf{p}\right)\right) \cdot r_{12} \\ & + \alpha \cdot 2 \cdot \Psi_2'\left(\left(\mathbf{r}_2^\top\left(\nabla u - \mathbf{p}\right)\right)^2 + \left(\mathbf{r}_2^\top\left(\nabla v - \mathbf{q}\right)\right)^2\right) \cdot \left(\mathbf{r}_2^\top\left(\nabla u - \mathbf{p}\right)\right) \cdot r_{22}, \end{aligned}$$

$$\begin{aligned}
F_{v_x} \;=\;& \alpha \cdot 2 \cdot \Psi_1' \left(\left(\mathbf{r}_1^\top (\nabla u - \mathbf{p}) \right)^2 + \left(\mathbf{r}_1^\top (\nabla v - \mathbf{q}) \right)^2 \right) \cdot \left(\mathbf{r}_1^\top (\nabla v - \mathbf{q}) \right) \cdot r_{11} \\
&+ \alpha \cdot 2 \cdot \Psi_2' \left(\left(\mathbf{r}_2^\top (\nabla u - \mathbf{p}) \right)^2 + \left(\mathbf{r}_2^\top (\nabla v - \mathbf{q}) \right)^2 \right) \cdot \left(\mathbf{r}_2^\top (\nabla v - \mathbf{q}) \right) \cdot r_{21}, \\
F_{v_y} \;=\;& \alpha \cdot 2 \cdot \Psi_1' \left(\left(\mathbf{r}_1^\top (\nabla u - \mathbf{p}) \right)^2 + \left(\mathbf{r}_1^\top (\nabla v - \mathbf{q}) \right)^2 \right) \cdot \left(\mathbf{r}_1^\top (\nabla v - \mathbf{q}) \right) \cdot r_{12} \\
&+ \alpha \cdot 2 \cdot \Psi_2' \left(\left(\mathbf{r}_2^\top (\nabla u - \mathbf{p}) \right)^2 + \left(\mathbf{r}_2^\top (\nabla v - \mathbf{q}) \right)^2 \right) \cdot \left(\mathbf{r}_2^\top (\nabla v - \mathbf{q}) \right) \cdot r_{22}, \\
F_{p_1} \;=\;& \alpha \cdot 2 \cdot \Psi_1' \left(\left(\mathbf{r}_1^\top (\nabla u - \mathbf{p}) \right)^2 + \left(\mathbf{r}_1^\top (\nabla v - \mathbf{q}) \right)^2 \right) \cdot \left(\mathbf{r}_1^\top (\nabla u - \mathbf{p}) \right) \cdot (-r_{11}) \\
&+ \alpha \cdot 2 \cdot \Psi_2' \left(\left(\mathbf{r}_2^\top (\nabla u - \mathbf{p}) \right)^2 + \left(\mathbf{r}_2^\top (\nabla v - \mathbf{q}) \right)^2 \right) \cdot \left(\mathbf{r}_2^\top (\nabla u - \mathbf{p}) \right) \cdot (-r_{21}), \\
F_{p_2} \;=\;& \alpha \cdot 2 \cdot \Psi_1' \left(\left(\mathbf{r}_1^\top (\nabla u - \mathbf{p}) \right)^2 + \left(\mathbf{r}_1^\top (\nabla v - \mathbf{q}) \right)^2 \right) \cdot \left(\mathbf{r}_1^\top (\nabla u - \mathbf{p}) \right) \cdot (-r_{12}) \\
&+ \alpha \cdot 2 \cdot \Psi_2' \left(\left(\mathbf{r}_2^\top (\nabla u - \mathbf{p}) \right)^2 + \left(\mathbf{r}_2^\top (\nabla v - \mathbf{q}) \right)^2 \right) \cdot \left(\mathbf{r}_2^\top (\nabla u - \mathbf{p}) \right) \cdot (-r_{22}), \\
F_{q_1} \;=\;& \alpha \cdot 2 \cdot \Psi_1' \left(\left(\mathbf{r}_1^\top (\nabla u - \mathbf{p}) \right)^2 + \left(\mathbf{r}_1^\top (\nabla v - \mathbf{q}) \right)^2 \right) \cdot \left(\mathbf{r}_1^\top (\nabla v - \mathbf{q}) \right) \cdot (-r_{11}) \\
&+ \alpha \cdot 2 \cdot \Psi_2' \left(\left(\mathbf{r}_2^\top (\nabla u - \mathbf{p}) \right)^2 + \left(\mathbf{r}_2^\top (\nabla v - \mathbf{q}) \right)^2 \right) \cdot \left(\mathbf{r}_2^\top (\nabla v - \mathbf{q}) \right) \cdot (-r_{21}), \\
F_{q_2} \;=\;& \alpha \cdot 2 \cdot \Psi_1' \left(\left(\mathbf{r}_1^\top (\nabla u - \mathbf{p}) \right)^2 + \left(\mathbf{r}_1^\top (\nabla v - \mathbf{q}) \right)^2 \right) \cdot \left(\mathbf{r}_1^\top (\nabla v - \mathbf{q}) \right) \cdot (-r_{12}) \\
&+ \alpha \cdot 2 \cdot \Psi_2' \left(\left(\mathbf{r}_2^\top (\nabla u - \mathbf{p}) \right)^2 + \left(\mathbf{r}_2^\top (\nabla v - \mathbf{q}) \right)^2 \right) \cdot \left(\mathbf{r}_2^\top (\nabla v - \mathbf{q}) \right) \cdot (-r_{22}), \\
F_{(\varphi_*)_x} \;=\;& \beta \cdot 2 \cdot \Psi_1' \left(|\mathcal{J}_\mathbf{p} \mathbf{r}_1|^2 + |\mathcal{J}_\mathbf{q} \mathbf{r}_1|^2 \right) \cdot \left(\mathbf{r}_1^\top \nabla \varphi_* \right) \cdot r_{11} \\
&+ \beta \cdot 2 \cdot \Psi_2' \left(|\mathcal{J}_\mathbf{p} \mathbf{r}_2|^2 + |\mathcal{J}_\mathbf{q} \mathbf{r}_2|^2 \right) \cdot \left(\mathbf{r}_2^\top \nabla \varphi_* \right) \cdot r_{21}, \\
F_{(\varphi_*)_y} \;=\;& \beta \cdot 2 \cdot \Psi_1' \left(|\mathcal{J}_\mathbf{p} \mathbf{r}_1|^2 + |\mathcal{J}_\mathbf{q} \mathbf{r}_1|^2 \right) \cdot \left(\mathbf{r}_1^\top \nabla \varphi_* \right) \cdot r_{12} \\
&+ \beta \cdot 2 \cdot \Psi_2' \left(|\mathcal{J}_\mathbf{p} \mathbf{r}_2|^2 + |\mathcal{J}_\mathbf{q} \mathbf{r}_2|^2 \right) \cdot \left(\mathbf{r}_2^\top \nabla \varphi_* \right) \cdot r_{22}. \qquad (4.47)
\end{aligned}$$

The resulting Euler-Lagrange equations are given by

$$0 = F_u - \alpha \cdot \operatorname{div} \left(D_{\mathrm{C,A}} \cdot (\nabla u - \mathbf{p}) \right), \qquad (4.48)$$

$$0 = F_v - \alpha \cdot \operatorname{div} \left(D_{\mathrm{C,A}} \cdot (\nabla v - \mathbf{q}) \right), \qquad (4.49)$$

$$0 = \alpha \cdot D_{\mathrm{C,A}} \cdot (\mathbf{p} - \nabla u) - \beta \cdot \mathbf{div} \left(\mathcal{J}_\mathbf{p} T_{\mathrm{C,A}} \right), \qquad (4.50)$$

$$0 = \alpha \cdot D_{\mathrm{C,A}} \cdot (\mathbf{q} - \nabla v) - \beta \cdot \mathbf{div} \left(\mathcal{J}_\mathbf{q} T_{\mathrm{C,A}} \right) \qquad (4.51)$$

where

$$D_{\mathrm{C,A}} := (\mathbf{r}_1 | \mathbf{r}_2) \begin{pmatrix} \Psi_1' & 0 \\ 0 & \Psi_2' \end{pmatrix} \begin{pmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \end{pmatrix} \qquad (4.52)$$

with $\Psi'_k := \Psi'_k((\mathbf{r}_k^\top(\nabla u - \mathbf{p}))^2 + (\mathbf{r}_k^\top(\nabla v - \mathbf{q}))^2)$, $k \in \{1,2\}$ and

$$T_{\mathrm{C,A}} := (\mathbf{r}_1|\mathbf{r}_2)\begin{pmatrix}\Psi'_1(|\mathcal{J}_\mathbf{p}\mathbf{r}_1|^2 + |\mathcal{J}_\mathbf{q}\mathbf{r}_1|^2) & 0 \\ 0 & \Psi'_2(|\mathcal{J}_\mathbf{p}\mathbf{r}_2|^2 + |\mathcal{J}_\mathbf{q}\mathbf{r}_2|^2)\end{pmatrix}\begin{pmatrix}\mathbf{r}_1^\top \\ \mathbf{r}_2^\top\end{pmatrix}. \tag{4.53}$$

The corresponding Neumann boundary conditions read

$$0 = \mathbf{n}^\top D_{\mathrm{C,A}} \cdot (\nabla u - \mathbf{p}), \qquad 0 = \mathbf{n}^\top D_{\mathrm{C,A}} \cdot (\nabla v - \mathbf{q}), \qquad 0 = \mathbf{n}^\top T_{\mathrm{C,A}} \cdot \nabla \varphi_*. \tag{4.54}$$

Again we can apply Stencil A.2 to discretize the divergence expressions containing the Jacobian. Stencils and their derivation for the coupling term are given in Table A.6 and Section B.4, respectively.

Compared to the isotropic regularization, the motion edges are more distinct in the anisotropic case as Figure 4.5 shows. In contrast to infimal convolution, we cannot see any affine flow fluctuations appear, while staircasing artifacts do not occur either.

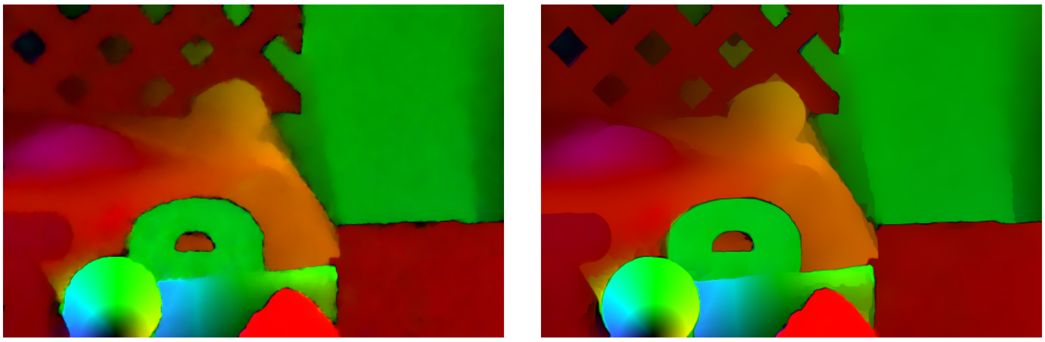

Figure 4.5: Comparison of the coupled regularizers. **Left:** Isotropic. **Right:** Anisotropic.

4.5 Summary

Several isotropic and anisotropic concepts realizing different ways of regularization have been introduced in this chapter. We presented diffusion processes that solely base on first order or second order derivatives. Both orders of derivatives, however, exhibited a bias to produce certain kinds of undesired image artifacts. Therefore, two following regularizers have been proposed combining derivatives of different order. While the first one still showed some artifacts with a first simple data set, the second one was able to provide very good estimates by getting rid of the unwanted effects. With an overview in Table 4.1 we conclude the subject of how to model optical flow computation using different kinds of first order and second order regularization and continue with their in-depth evaluation in the following chapter.

Type	**Isotropic Regularization**		**Anisotropic Regularization**	
First Order Derivatives	$R_{\mathrm{I,I}}$ [15]	$\Psi\Big(\sum_{i=1}^{2}\lvert\nabla\mu_i\rvert^2\Big)$	$\sum_{j=1}^{2}\Psi_j\Big(\sum_{i=1}^{2}(\mathbf{r}_j^\top\nabla\mu_i)^2\Big)$	$R_{\mathrm{I,A}}$ [21]
Second Order Derivatives	$R_{\mathrm{II,I}}$ [23]	$\Psi\Big(\sum_{i=1}^{2}\lVert\mathcal{H}_{\mu_i}\rVert_{\mathrm{F}}^2\Big)$	$\sum_{j=1}^{2}\Psi_j\Big(\sum_{i=1}^{2}\lvert\mathcal{H}_{\mu_i}\mathbf{r}_j\rvert^2\Big)$	$R_{\mathrm{II,A}}$
Derivatives Combined	$R_{\mathrm{III,I}}$ [22]	$\Psi\Big(\sum_{i=1}^{2}\lvert\nabla\mu_{i,1}\rvert^2\Big)$ $+\beta\Psi\Big(\sum_{i=1}^{2}\lVert\mathcal{H}_{\mu_{i,2}}\rVert_{\mathrm{F}}^2\Big)$	$\sum_{j=1}^{2}\Psi_j\Big(\sum_{i=1}^{2}(\mathbf{r}_j^\top\nabla\mu_{i,1})^2\Big)$ $+\beta\sum_{j=1}^{2}\Psi_j\Big(\sum_{i=1}^{2}\lvert\mathcal{H}_{\mu_{i,2}}\mathbf{r}_j\rvert^2\Big)$	$R_{\mathrm{III,A}}$
Derivatives Coupled	$R_{\mathrm{IV,I}}$ [25]	$\Psi\Big(\sum_{i=1}^{2}\lvert\nabla\mu_i-\boldsymbol{\varphi}_i\rvert^2\Big)$ $+\beta\Psi\Big(\sum_{i=1}^{2}\lVert\mathcal{J}_{\boldsymbol{\varphi}_i}\rVert_{\mathrm{F}}^2\Big)$	$\sum_{j=1}^{2}\Psi_j\Big(\sum_{i=1}^{2}(\mathbf{r}_j^\top(\nabla\mu_i-\boldsymbol{\varphi}_i))^2\Big)$ $+\beta\sum_{j=1}^{2}\Psi_j\Big(\sum_{i=1}^{2}\lvert\mathcal{J}_{\boldsymbol{\varphi}_i}\mathbf{r}_j\rvert^2\Big)$	$R_{\mathrm{IV,A}}$ [26]

Table 4.1: Overview of the introduced regularizers, each in its isotropic and anisotropic variant. In this context we use the notation $\mu_1 := u, \mu_2 := v, \boldsymbol{\varphi}_1 := \mathbf{p}$ and $\boldsymbol{\varphi}_2 := \mathbf{q}$.

5 Experimental Evaluation

Previously, we explained several aspects of variational optical flow computation that range from modeling and realizing certain assumptions on the initial input image data and the final estimation to solving the linearized equation system numerically (Chapter 3). Furthermore, various concepts to regularize the motion field have been introduced (Chapter 4). To compare their capabilities for optical flow computation appropriately we use a common set of input images, parameter values and software. The computation itself is performed in a modified framework implementing all the different regularization terms, that was originally provided by the *Intelligent Systems* department.
Before we start with our evaluation in this chapter, we give an insight about the hard- and software environment running the framework, chosen parameter values, optimization strategy for certain parameters and characteristics of the selected data sets.

Hardware and Software Environment. The optical flow computation was initially performed on a standard Windows 7 x64 home PC. Since the optimization was quite a lengthy process (20536 optimization steps for the final results plus many steps of repeated runs due to bad local minima), a laptop was added later during optimization to perform some estimations in parallel. PC and laptop contained relatively current hardware at evaluation time as we can see in Table 5.1. While both machines were using Windows 7 64-bit as the operating system, a virtual machine needed to be set up, since the framework only run in a Linux environment. We chose Xubuntu 16.04.1 as a lightweight Linux operating system.

Parameter Details. During our explanation of optical flow and its various regularization terms, many additional parameters have been introduced. Since it is impossible for us to

	PC	Laptop
CPU	Intel Core i7-4770K @ 3.5 GHz	Intel Core i7-5600U @ 2.6 GHz
Mainboard	Gigabyte Z87X-UD4H	Lenovo 20BWS2KM00
Chipset	Intel Z87	Intel Broadwell-U PCH-LP
RAM	16 GB DDR3-1600/PC3-12800	12 GB DDR3-1600/PC3-12800
SSD	256 GB, Samsung 840 Pro	512 GB, LiteOn LCH-512V2S
Host OS	Windows 7 Professional 64-bit	Windows 7 Professional 64-bit
Guest OS	Xubuntu 16.04.1 64-bit	Xubuntu 16.04.1 64-bit

Table 5.1: Hardware and software used for evaluation purposes.

optimize them all jointly within a reasonable amount of time, we restricted ourselves to only optimize the regularization parameters α and β with respect to the AEE. For all the other parameters we chose fixed values, which can be learned from Table 5.2.
To find the best average values for α (and β in case of combined and coupled regularization) per data set, we used a *Downhill Simplex* strategy [43], which is not further explained here. We only want to point out that this method highly depends on suitable initial values to start the optimization from. Otherwise it might get trapped in bad local minima. This behavior caused us to restart some optimization runs with other more appropriate initializations.

Penalizer (Data Term)	**Normalization**	**CRT Neighborhood**	**Presmoothing**	**Relaxation**
$\epsilon = 0.00003$	$\zeta = 0.01$	$\kappa = 9$	$\sigma = 0.3$	$\omega = 1.9$
Penalizer (Smoothness Term)	**Warping Levels**	**Refinement**	**Inner Iterations**	**Relaxation Steps**
$\epsilon = 0.01$	$k_{\max} = 200$	$\eta = 0.95$	$l_{\max} = 10$	$m_{\max} = 5$

Table 5.2: Fixed parameters employed during evaluation.

5.1 Data Sets

A detailed evaluation covering various realistic scenarios stands and falls with the employed test data. Therefore, a proper selection of a wide range of data sets is necessary. We choose specific test image sequences that contain different types of motion (translational, divergent, rotational), large displacements, complex object structures and several lighting effects as well as shadows. To compare the optical flow estimations to the real occurring motion using an error metric (see Section 2.5), ground truth images for each image sequence are needed. The well-known and widely-used image databases of the Middlebury benchmark [33], the Sintel benchmark [44] and the KITTI benchmark [7, 45] are collections of suitable image data and ground truth flow where we take our test images from.

5.1.1 Middlebury Benchmark

Baker et al. [33] published their evaluation data as one of the first collections to present the limits and constraints of current optical flow methods at that time (2007) and to point out in which direction next generation techniques should develop. The data set consists of real world scenarios as well as synthetically rendered scenes with more realistic textures. While different types of motion occur in these mostly rigid scenes, illumination changes only appear slightly compared to other data sets causing today's methods to produce very good results already. Nevertheless, the Middlebury benchmark data set can show first tendencies of estimation quality, which is a good start for our evaluation. Table 5.3 and Figure 5.1 give an overview about the chosen Middlebury test images and their properties.

Sequence	Frame #	Properties	Resolution
Dimetrodon	10 & 11	Real world scene with a toy. Dominant colors. Mostly translational left oriented motion.	584 × 388
Hydrangea	10 & 11	Real world scene with a flower. Translational motion in right direction and clockwise oriented rotational motion.	584 × 388
RubberWhale	10 & 11	Real world scene with many differently colored objects. Some objects exhibit translational motion. Other objects move in a rotational way.	584 × 388
Grove2	10 & 11	Synthetic scene with trees and stones. Environmental motion oriented in left direction. Front tree moves towards bottom left corner.	640 × 480
Grove3	10 & 11	Synthetic scene with trees and stones. Divergent motion. Front objects move in right and bottom direction. Back objects move upwards.	640 × 480
Urban2	10 & 11	Synthetic scene with buildings. Global rotational motion.	640 × 480
Urban3	10 & 11	Synthetic scene with buildings. Divergent motion. Front buildings move in bottom direction. Background moves upwards.	640 × 480
Venus	10 & 11	Synthetic setup with heavily textured paintings. Translational motion in left and right direction.	420 × 380

Table 5.3: Overview of the selected Middlebury test image sequences and their properties.

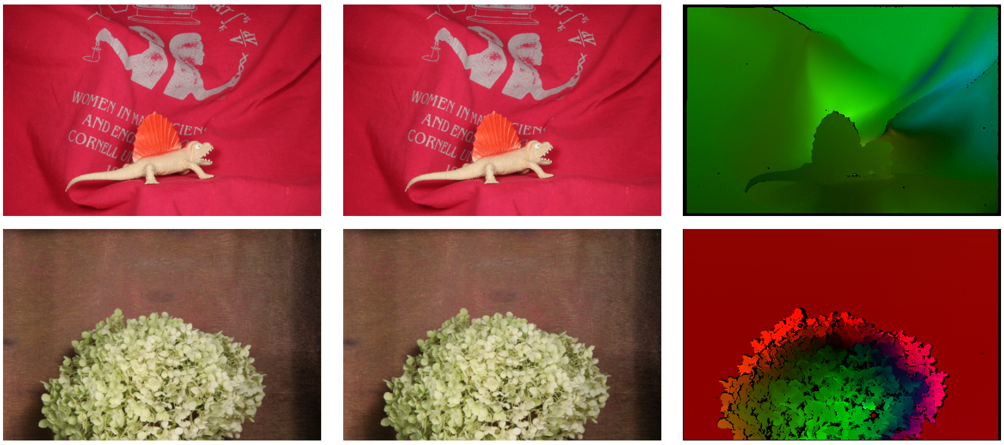

Figure 5.1: Overview of the selected test image sequences with corresponding ground truth.

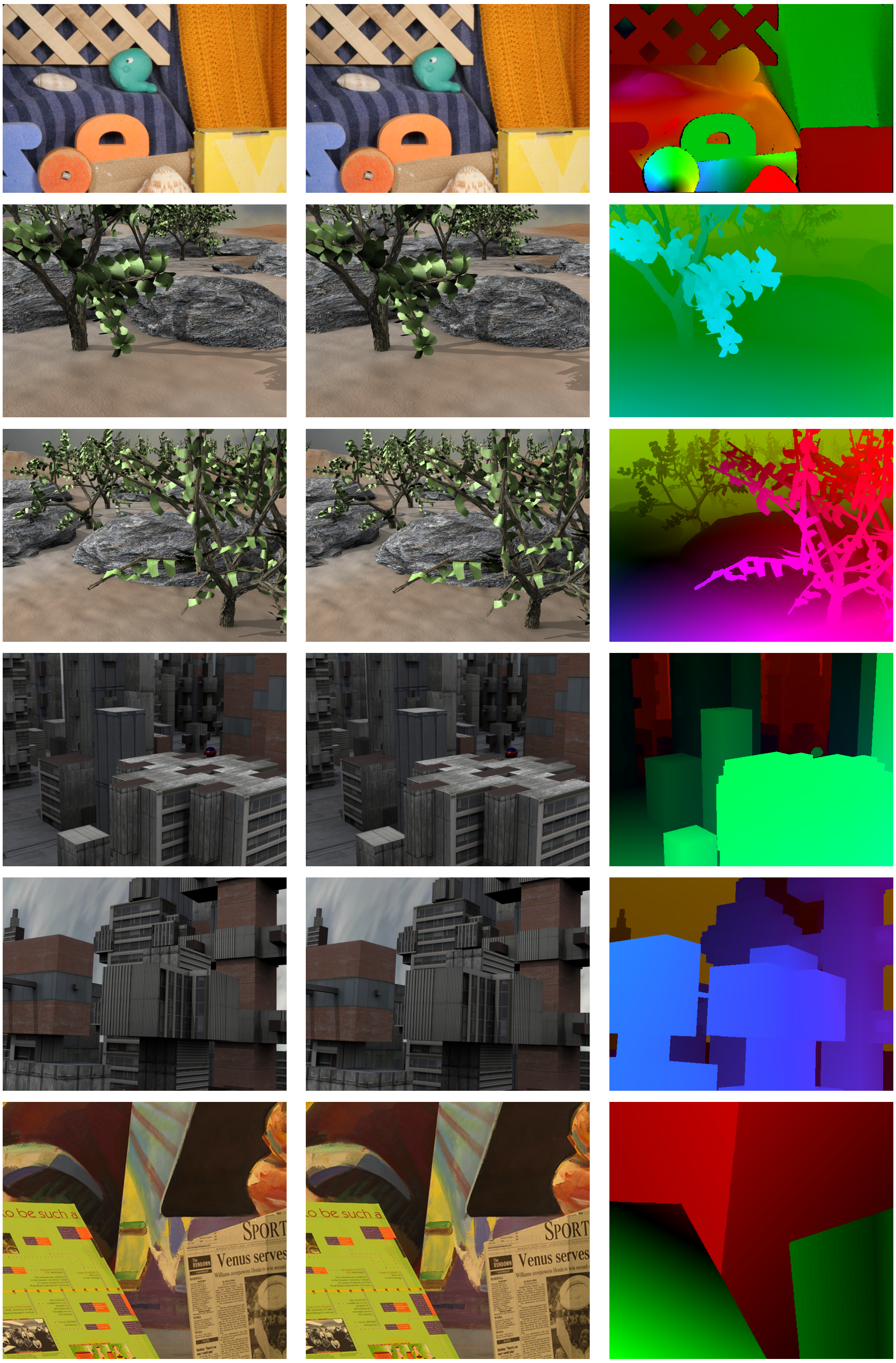

Figure 5.1: Overview of the selected test image sequences with corresponding ground truth (*continued*). **Left to Right:** Reference frame, subsequent frame and ground truth. **Top to Bottom:** Dimetrodon, Hydrangea, RubberWhale, Grove2, Grove3, Urban2, Urban3 and Venus.

5.1.2 Sintel Benchmark

Displacements of objects in the Middlebury data set were considered to be quite large back then (up to 35 pixels per frame (ppf) in the synthetic scenes), but state-of-the-art methods can even handle larger and faster motion. Actual real world scenarios do not only contain different independent motion, but also incorporate more complex materials with reflecting surfaces, shadows, varying illumination and natural shading effects, and a higher degree of general complexity. Butler et al. [44] respect all these characteristics in their synthetic benchmark data set (2012) based on the open source 3D animated short film Sintel. We will use test images from their clean rendering pass that resembles realistic scenes better with motion partly larger than 100 ppf, but omits image degradation such as blurring and atmospheric effects. The selected Sintel test images and some of their characteristics are presented in Table 5.4 and Figure 5.2.

Sequence	Frame #	Properties	Resolution
Alley1	18 & 19	Calm scene with slow camera and object motion. Similar color impression of different objects. Slightly altering shading.	1024 × 436
Ambush7	39 & 40	Scene with mostly steady background (snow). Moving objects clearly separated from background. Large motion. Varying shading.	1024 × 436
Bamboo2	35 & 36	Scene in a bamboo forest with many independently moving (small) objects. Camera motion. Dominating color. Shading changes slightly.	1024 × 436
Cave4	34 & 35	Drab scene in a cave. Mostly grayish and dark colors. Object and camera motion. Varying shading.	1024 × 436
Market2	18 & 19	Scene at a busy market with many objects. Large and small motion. Slight camera motion. Numerous objects with brownish surface.	1024 × 436
Market6	10 & 11	Fast scene at a market. Large camera motion. Similar overall color impression. Changing shading.	1024 × 436
Shaman2	36 & 37	Rather dark scene with some bright reflections. Small motion of many objects (hairs of the beard).	1024 × 436
Sleeping1	12 & 13	Scene with many shadows. Globally small rotational and divergent motion caused by zooming camera. Motion of objects almost not present.	1024 × 436
Temple3	29 & 30	Very hasty scene with large motion of all objects. Moving background. Shading alters drastically.	1024 × 436

Table 5.4: Overview of the selected Sintel test image sequences and their properties.

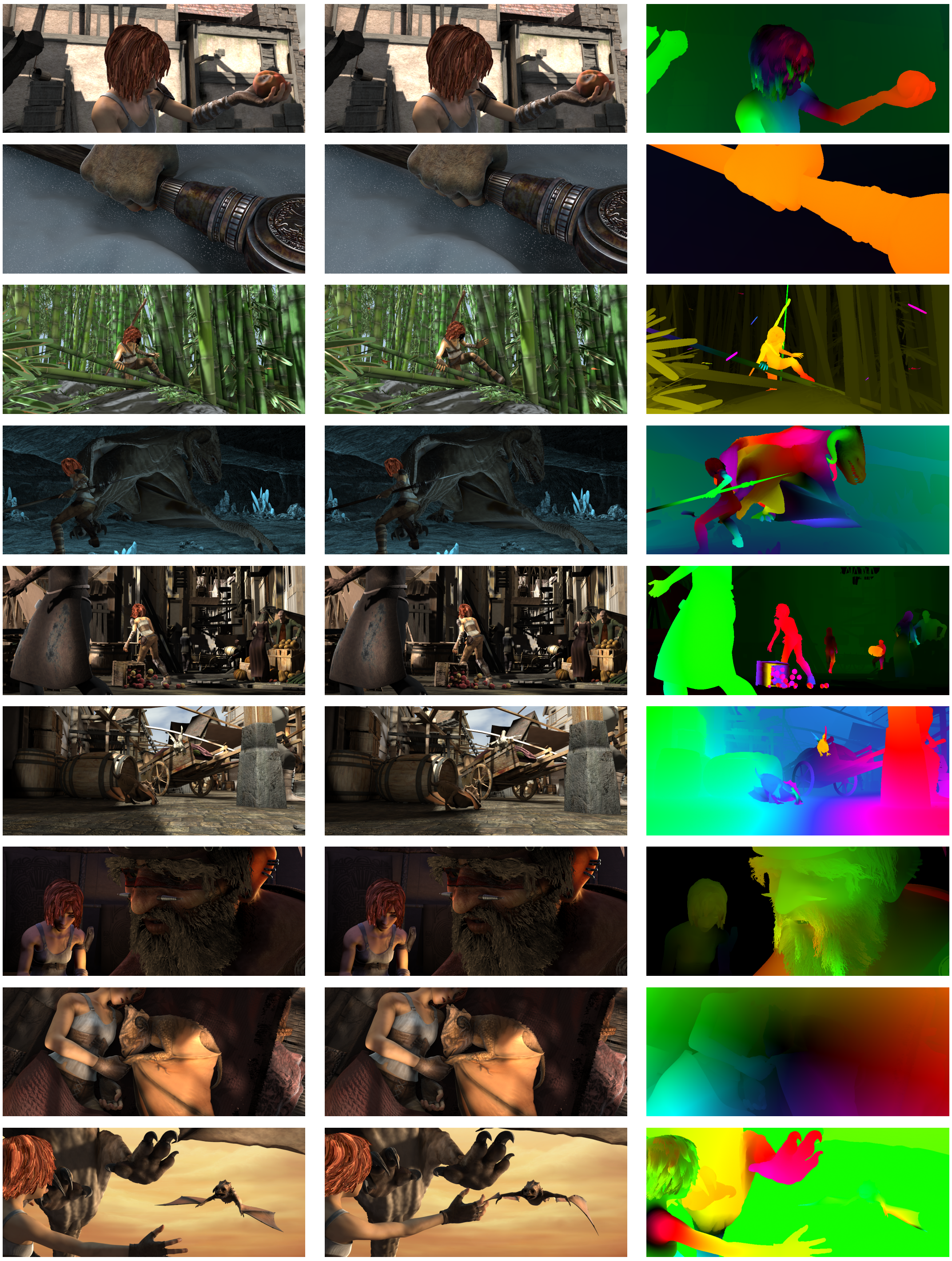

Figure 5.2: Overview of the selected test image sequences with corresponding ground truth. **Left to Right:** Reference frame, subsequent frame and ground truth. **Top to Bottom:** Alley1, Ambush7, Bamboo2, Cave4, Market2, Market6, Shaman2, Sleeping1 and Temple3.

5.1.3 KITTI Benchmark

While we refrain from using Sintel's final rendering pass that includes motion and defocus blur as well as other atmospheric effects, a third data set is added to our evaluation captured in the real world where such effects occur naturally. Menze et al. [45] chose to gather their test data in a realistic application scenario (autonomous driving) by driving around in car on public streets. The car was equipped with different cameras to take images and a 3D laser scanner to obtain motion information from independent objects and thus create the ground truth data. We decide to use the latest KITTI benchmark data set (2015) since - compared to the data set from 2012 [7] - it contains more challenging sequences and ground truth data for non-rigid scenes with independently moving objects. Please note that ground truth has to be measured for these scenes since they are not staged or synthetic and hence only exhibits a density of around 50% in average. Again, the chosen KITTI test images are shown in Table 5.5 and Figure 5.3.

Sequence	Frame #	Properties	Resolution
14	10 & 11	Rather bright scene caused by sunlight. Test car slowly stopping at traffic lights. Other vehicles moving over crossroads.	1242 × 375
29	10 & 11	Large part of the scene covered in shadows casted by trees. Test car following a straight road with normal speed. Other vehicles moving in front or approaching on opposing lane.	1242 × 375
46	10 & 11	Bright scene. Test car standing still at traffic lights. Stationary shadow casted on car crossing the lane.	1242 × 375
86	10 & 11	Test car moving slowly in a residential area. Houses cause many shadows. Other car crossing the street. White and brownish colors dominating the scene.	1242 × 375
120	10 & 11	Bright scene with public street, houses and parking cars casting some shadows. Other vehicles moving towards the test car or driving in the same direction.	1242 × 375
144	10 & 11	Scene in daylight. Subway causes large shadow. Ground truth for subway not available.	1242 × 375
171	10 & 11	Scene with strongly reflecting building. Street mostly covered in shadows. Test car and other vehicle in front curving the crossroads. Reflections cause some disturbances on the camera lense.	1242 × 375

Table 5.5: Overview of the selected KITTI test image sequences and their properties.

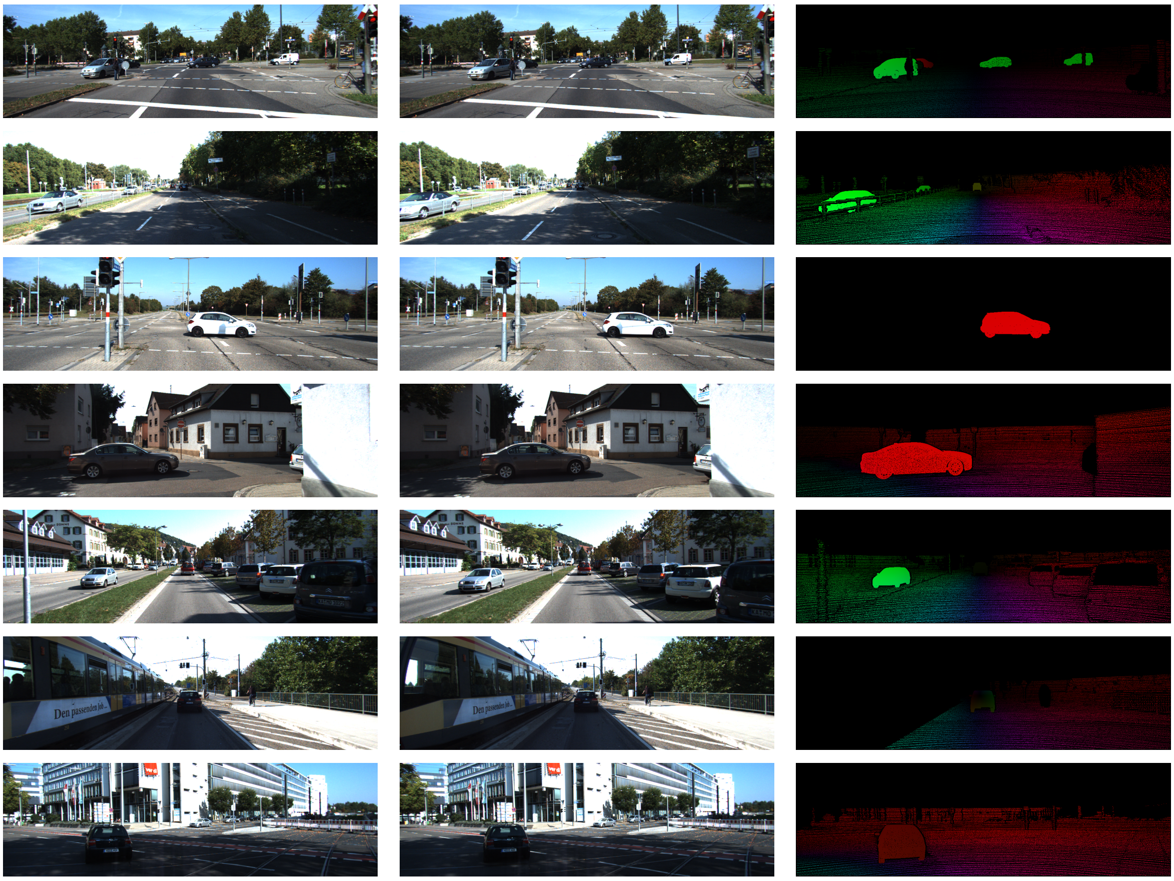

Figure 5.3: Overview of the selected test image sequences with corresponding ground truth (around 50% density). **Left to Right:** Reference frame, subsequent frame and ground truth. **Top to Bottom:** Sequence 14, 29, 46, 86, 120, 144 and 171.

5.2 Comparison of the Regularizers

With the just introduced data sets at hand the evaluation of the eight different regularization terms (see Table 4.1 for recap) is now ready to begin with. In addition, we will make use of the gray value/brightness constancy assumption (BCA), the gradient constancy assumption (GCA) and the complete rank transform (CRT) (see summary in Table 3.1) combined with the various data term modifications given in Table 3.2 to study arising differences in the final motion estimations. To penalize deviations in the data term we continuously use the Charbonnier function Ψ_{CH}. As mentioned before the regularization weights α and β are optimized with respect to the AEE. In the following we will examine the final results for each data set separately, starting with the Middlebury data set as the least challenging one.

5.2.1 Results - Middlebury

Our first evaluation using the well-established Middlebury data set provides some initial insights about optical flow estimation based on different constancy assumptions and regularization approaches. The results in form of the AEE and optimized parameter values are carried together in Table 5.6. For one particular regularization strategy the AEE is given for each individual image sequence as well as averaged over all sequences. To emphasize the impact of different constancy assumptions the regularization strategies are compared for every assumption separately, where we highlight the best result in green and color the worst one red.

Table 5.6 reveals a common error development that appears no matter what constancy assumption we apply: (1) Second order regularization provides worse results than first order regularization in all cases. (2) Combined regularization has a tendency to favor solutions which are very similar to second order regularization. (3) Coupled regularization gives either better results than first order regularization or is at least very close to it (GCA case). (4) In most of the cases the anisotropic variant of a certain regularizer yields better results than its isotropic counterpart (except BCA first order and coupled regularization).

If we have closer look on the average errors now, we can see that coupled isotropic regularization performs best in the BCA case, while second order isotropic regularization gives the worst result. When it comes to the GCA, first order anisotropic regularization has the lowest error. But it does not only give the best estimation with respect to the GCA. In fact, GCA first order anisotropic regularization yields the best estimation of the entire Middlebury data set. Coupled anisotropic regularization gives the most accurate estimation with the CRT, but it cannot beat the best results of the GCA. The CRT even provides the worst overall result (second order isotropic). While anisotropic regularization sharpens motion edges and thus helps to improve the estimation quality, it also causes undesired image artifacts and aberrations sometimes (see Figure 5.4).
The simplicity of the Middlebury data set with its rigid scenes, sparse illumination changes (too many constraints of the CRT may even worsen the results slightly) and a low degree of egomotion of the camera (affine motion not that much present) could be a reason why first order regularization together with the GCA dominates over the other regularization strategies. Nevertheless, all the regularization strategies already perform very well on the Middlebury data set with respect to the AEE. The difference between the overall best and worst result is relatively small compared to the following data sets.

Figure 5.5 gives a final overview of differently regularized motion estimation with the GCA of our running example: the RubberWhale sequence. Figure 5.6, Figure 5.7, Figure 5.8, and Figure 5.9 show a direct comparison between regularization with the GCA and the CRT using the Urban3 sequence.
The Urban3 motion visualizations based on the GCA and the CRT confirm the numerical result. While the GCA achieves more homogeneous flows and accurate edges, the CRT suffers from blurred motion boundaries (especially in the second order and the combined case) and a wrong flow field at the front wall of the central building. Both drawbacks are also clearly visible in the brighter areas of the difference images.

Middlebury		First Order		Second Order		Combined Regularization		Coupled Regularization	
		Iso	Aniso	Iso	Aniso	Iso	Aniso	Iso	Aniso
BCA AEE	Dimetro.	0.1389	0.1406	0.1623	0.1585	0.1662	0.1589	0.1388	0.1409
	Hydrangea	0.2033	0.1929	0.2513	0.2456	0.2554	0.2469	0.1995	0.2014
	RubWha.	0.1483	0.1286	0.1774	0.1740	0.1725	0.1724	0.1417	0.1335
	Grove2	0.1294	0.1172	0.1617	0.1458	0.1608	0.1459	0.1297	0.1221
	Grove3	0.5260	0.5319	0.5824	0.5468	0.5592	0.5376	0.5261	0.5268
	Urban2	0.2986	0.3849	0.3562	0.4178	0.3537	0.4263	0.2773	0.4344
	Urban3	0.3663	0.3170	0.5448	0.5141	0.5375	0.5070	0.3831	0.3212
	Venus	0.2877	0.3344	0.5228	0.4914	0.5082	0.4764	0.2775	0.2976
BCA	**AVG AEE**	0.2623	0.2684	0.3448	0.3368	0.3392	0.3339	0.2592	0.2722
	α	0.0057	0.0141	0.0075	0.0040	0.5638	0.0805	0.0066	0.0100
	β	–	–	–	–	0.0038	0.0026	3.5835	0.3762
GCA AEE	Dimetro.	0.0973	0.0972	0.1129	0.1009	0.1086	0.1012	0.0973	0.0971
	Hydrangea	0.1423	0.1354	0.1586	0.1479	0.1594	0.1480	0.1425	0.1454
	RubWha.	0.0794	0.0683	0.1012	0.0891	0.1016	0.0891	0.0791	0.0693
	Grove2	0.1413	0.1198	0.1708	0.1559	0.1705	0.1560	0.1416	0.1286
	Grove3	0.5860	0.5319	0.6312	0.6087	0.6365	0.6083	0.5786	0.5379
	Urban2	0.2983	0.2218	0.3662	0.3270	0.3629	0.3281	0.3059	0.2675
	Urban3	0.4190	0.3153	0.5723	0.5305	0.5783	0.5289	0.3992	0.2918
	Venus	0.2975	0.2684	0.3765	0.3375	0.3722	0.3376	0.2853	0.2379
GCA	**AVG AEE**	0.2576	0.2198	0.3112	0.2872	0.3113	0.2872	0.2537	0.2219
	α	0.0056	0.0095	0.0063	0.0036	2.7221	2.0827	0.0059	0.0092
	β	–	–	–	–	0.0050	0.0025	1.0535	0.9460
CRT AEE	Dimetro.	0.1002	0.0999	0.1162	0.1046	0.1116	0.1028	0.1001	0.0984
	Hydrangea	0.1427	0.1371	0.1654	0.1556	0.1654	0.1561	0.1422	0.1425
	RubWha.	0.0864	0.0763	0.1110	0.1002	0.1108	0.1004	0.0856	0.0764
	Grove2	0.1436	0.1243	0.1958	0.1772	0.1918	0.1758	0.1430	0.1287
	Grove3	0.5619	0.5417	0.6252	0.6084	0.6325	0.6138	0.5458	0.5190
	Urban2	0.3190	0.2597	0.4313	0.4008	0.4190	0.3843	0.3105	0.2967
	Urban3	0.5393	0.4677	0.8733	0.8415	0.8540	0.8338	0.5295	0.4481
	Venus	0.2890	0.2757	0.4232	0.3771	0.4133	0.3726	0.2643	0.2296
CRT	**AVG AEE**	0.2728	0.2478	0.3677	0.3456	0.3623	0.3425	0.2651	0.2424
	α	0.0095	0.0188	0.0106	0.0063	1.0291	0.5141	0.0095	0.0134
	β	–	–	–	–	0.0088	0.0049	3.6995	1.6727

Table 5.6: Complete set of error data with corresponding weights achieved during evaluation of the Middlebury data set.

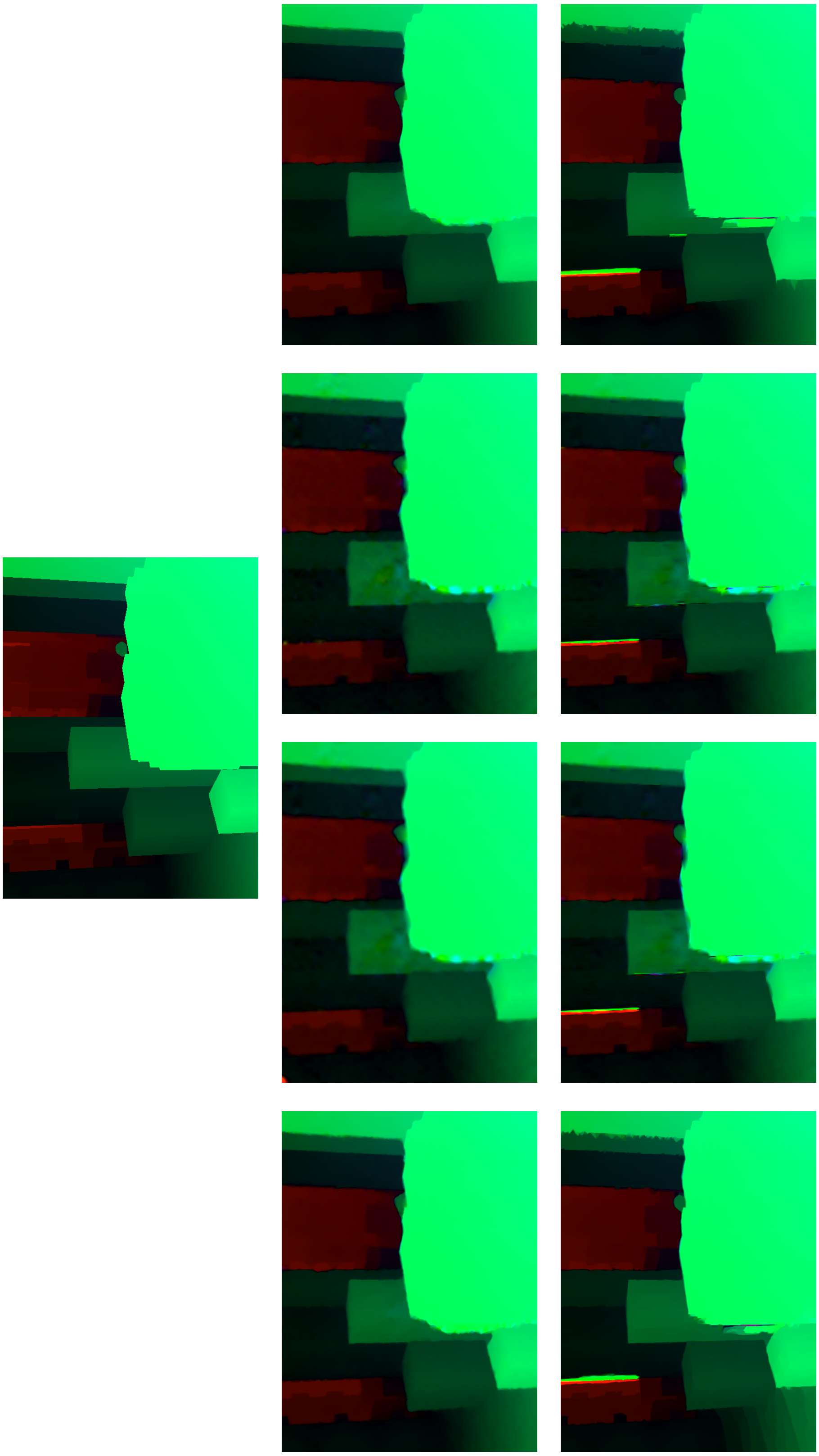

Figure 5.4: Artifacts and aberrations at motion edges (especially first order and coupled regularization) with anisotropic regularization appearing in the Urban2 sequence. **First row:** Ground truth image. **Top to Bottom:** Isotropic and anisotropic regularization. **Left to Right:** First order, second order, combined and coupled regularization.

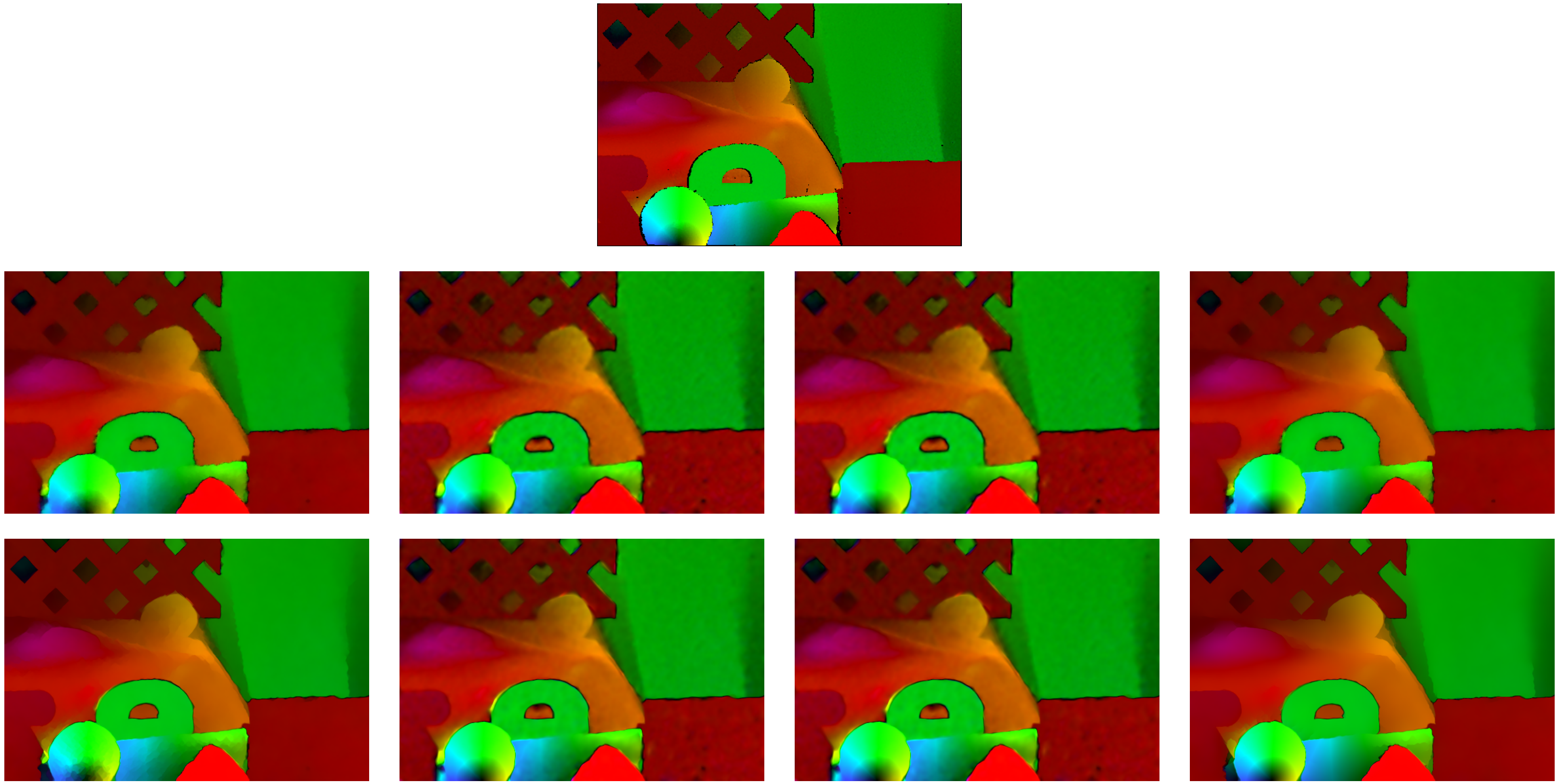

Figure 5.5: Best results for the RubberWhale sequence achieved using the GCA. **First row:** Ground truth image. **Top to Bottom:** Isotropic and anisotropic regularization. **Left to Right:** First order, second order, combined and coupled regularization.

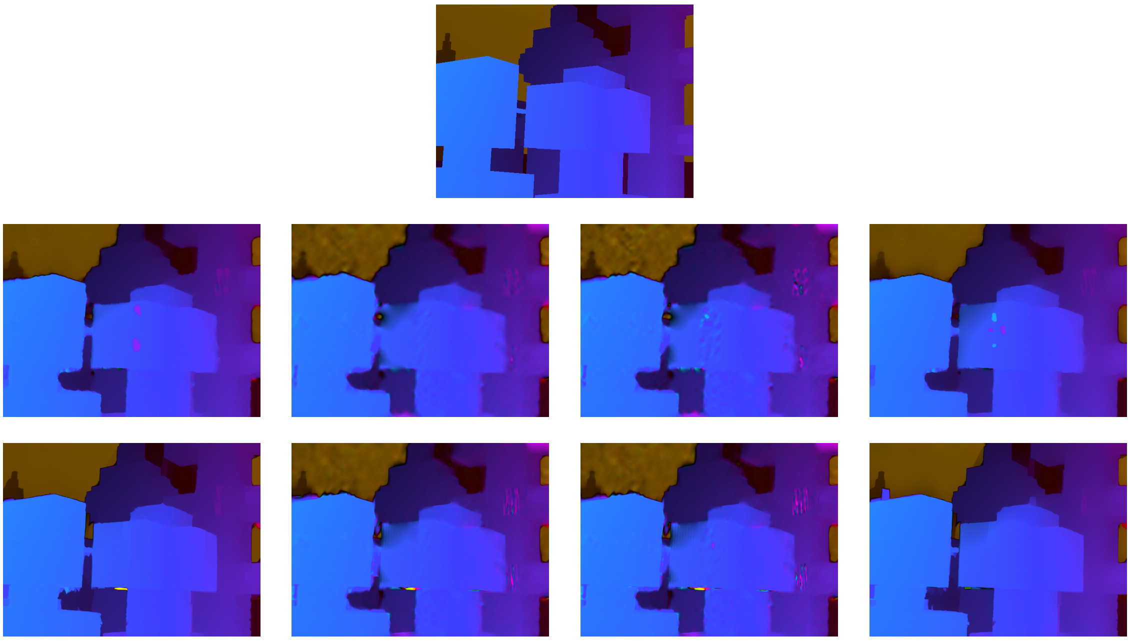

Figure 5.6: Results for the Urban3 sequence using different regularizers and the GCA. **First row:** Ground truth. **Top to Bottom:** Isotropic and anisotropic regularization. **Left to Right:** First order, second order, combined and coupled regularization.

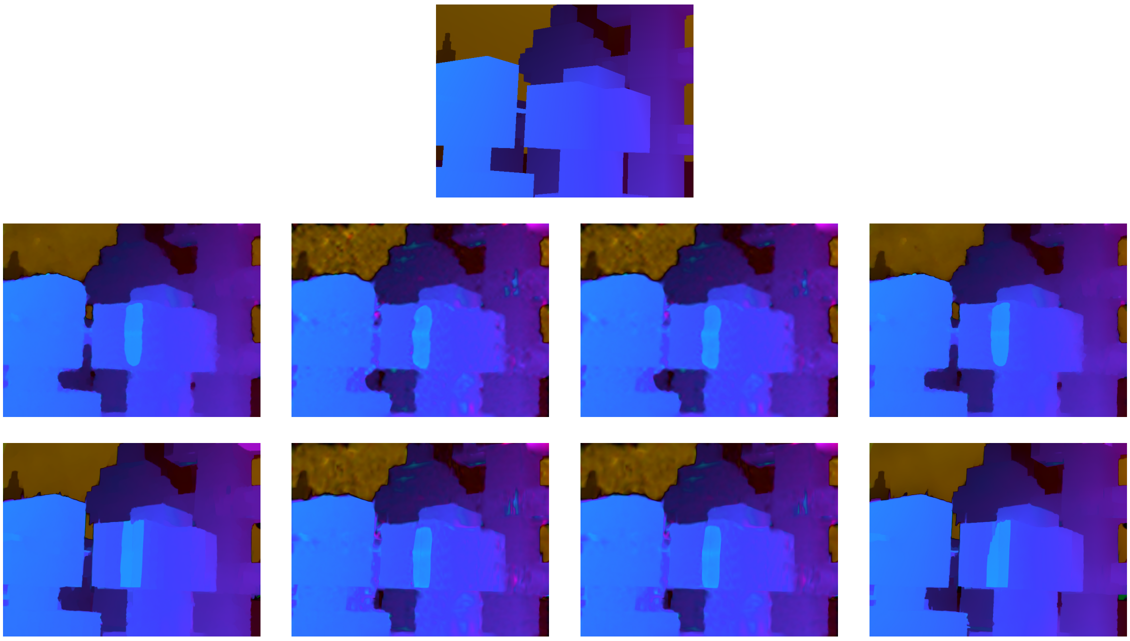

Figure 5.7: Results for the Urban3 sequence using different regularizers and the CRT. **First row:** Ground truth. **Top to Bottom:** Isotropic and anisotropic regularization. **Left to Right:** First order, second order, combined and coupled regularization.

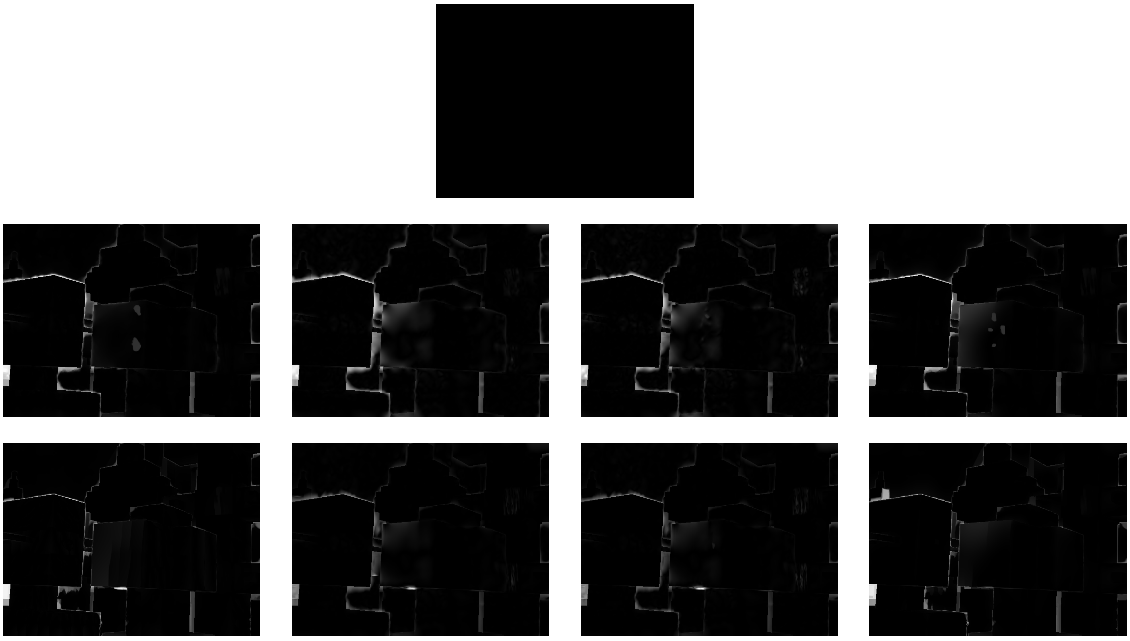

Figure 5.8: Differences to ground truth of corresponding regularizers using the GCA. **First row:** No difference. **Top to Bottom:** Isotropic and anisotropic regularization. **Left to Right:** First order, second order, combined and coupled regularization.

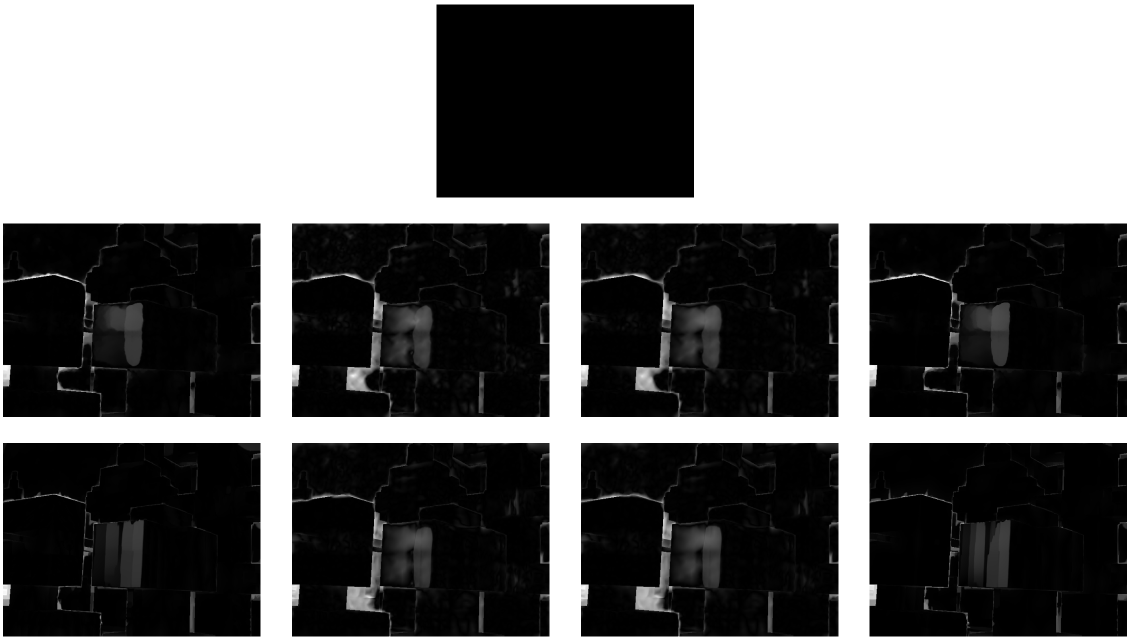

Figure 5.9: Differences to ground truth of corresponding regularizers using the CRT. **First row:** No difference. **Top to Bottom:** Isotropic and anisotropic regularization. **Left to Right:** First order, second order, combined and coupled regularization.

5.2.2 Results - Sintel

Compared to the Middlebury test images our current data set consists of more complex scenes that, among other things, include large motion and changing illumination. If we compare the average errors from the BCA over the GCA to the CRT, a decreasing error in each of the regularizers can be observed. The Ambush7 sequence contains varying shading and thus, despite having simple translational motion, can clearly show the advantage of more advanced constancy assumptions. Figure 5.10 contrasts the best result using the BCA (first order isotropic) with the lowest error provided by the CRT (coupled anisotropic). The BCA does not allow to distinguish motion of the objects from the mostly steady background.

After highlighting that the CRT works best on Sintel data set, let us now discuss the average results of the separate regularizers. Again, the tendency of the AEE is the same regardless of the selected constancy assumption as presented in Table 5.7: (1) In all cases anisotropic regularization performs worse than its isotropic counterpart (contrary to previous Middlebury observations). (2) Second order anisotropic regularization always provides the worst estimations. (3) Combined regularization yields better solutions than second order, but worse than first order regularization. (4) The best results are achieved with coupled regularization (except the GCA case, which is discussed later yet).

Considering individual results now, the common impression changes a little. As already pointed out the BCA cannot handle shadows very well (see results of the Ambush7 sequence). If we use the GCA or the CRT instead, low errors can be achieved in almost all sequences - unless in the Market6 and the Temple3 sequence as considerably higher errors indicate. We focus on these particular sequences now.
The Market6 sequence is a fast and complex scene with large egomotion of the camera. Such a large egomotion typically cause many occlusions or rather unmatched regions, especially at the image boundaries. Occlusions appear if a pixel of an object is visible in one of the frames, but hidden in the other one due to objects covering other objects or objects simply moving out of the camera view. Since correspondences cannot be found for occluded pixels, the error will increase in such regions. This may explain the higher error compared to other sequences. Another aspect of this sequence that we want to mention is the strong affine motion caused by the egomotion of the camera. In fact, the Market6 sequence is the only sequence where second order and combined regularization perform better than first order regularization if the GCA or the CRT is applied (values colored orange). The visualizations in Figure 5.11 and Figure 5.12 show, however, that the motion of the chicken is not estimated correctly in any of the GCA results (neither in the CRT case having similar visualizations). Additionally, occlusions cause bad estimations at the right and bottom left boundary.
Of all the chosen Sintel test images the Temple3 sequence features the greatest challenge for motion estimation. Large motion of almost every object in the scene, changing illumination and occlusion cause the highest error values in the entire data set. Without the Temple3 sequence GCA coupled regularization would even yield better average values than GCA first order regularization. While first order and coupled regularization provide more accurate estimations than second order and combined regularization, Figure 5.13 and Figure 5.14 indicate that their anisotropic variants lead to more appealing results in some areas (arm and hair), but they also cause a large region with many errors (top right: dragon's wing).

Sintel		First Order		Second Order		Combined Regularization		Coupled Regularization	
		Iso	Aniso	Iso	Aniso	Iso	Aniso	Iso	Aniso
BCA AEE	Alley1	0.1917	0.1684	0.2578	0.2572	0.3230	0.3148	0.1854	0.1688
	Ambush7	7.0886	7.5790	8.7514	10.618	10.224	10.048	7.7128	8.1040
	Bamboo2	0.7245	0.6734	0.8824	0.9137	0.8614	0.9278	0.6963	0.6830
	Cave4	0.8558	0.7574	1.1320	1.1965	1.0838	1.1467	0.8269	0.7686
	Market2	0.9670	0.8987	1.7854	1.7905	1.6686	1.3809	1.0183	0.9555
	Market6	9.5056	9.6748	11.380	12.709	11.022	9.0353	7.1126	7.8038
	Shaman2	0.1529	0.1429	0.2178	0.2088	0.2353	0.3154	0.1583	0.1476
	Sleeping1	0.1032	0.1187	0.1297	0.1246	0.1326	0.1599	0.0941	0.0847
	Temple3	21.876	22.526	35.629	34.988	27.848	31.937	23.271	24.525
BCA	**AVG AEE**	4.6072	4.7266	6.6851	6.9784	5.9332	6.1407	4.5640	4.8045
	α	0.0126	0.0220	0.0157	0.0095	0.0346	0.0322	0.0096	0.0134
	β	–	–	–	–	0.0090	0.0014	1.3287	2.9921
GCA AEE	Alley1	0.1521	0.1184	0.2317	0.1695	0.2350	0.4044	0.2167	0.1150
	Ambush7	2.1075	2.3352	3.5848	4.7897	3.7365	4.1184	1.9609	1.4772
	Bamboo2	0.7988	0.7018	0.8226	0.8302	0.8248	1.0052	0.7185	0.6846
	Cave4	0.9218	0.8331	0.8902	0.8683	0.9156	1.1041	0.7536	0.7964
	Market2	1.1021	0.8713	1.1507	1.0894	1.2863	1.4387	1.0064	0.8466
	Market6	9.7516	10.114	7.9381	10.591	8.1467	7.8051	5.6014	8.1108
	Shaman2	0.1220	0.1132	0.1992	0.1465	0.2066	0.3390	0.1173	0.1141
	Sleeping1	0.0785	0.0988	0.1194	0.0857	0.1237	0.2034	0.0665	0.0581
	Temple3	17.481	17.343	35.565	34.322	28.921	33.966	23.553	23.037
GCA	**AVG AEE**	3.6128	3.6143	5.6112	5.8768	4.9329	5.5983	3.7772	3.9155
	α	0.0144	0.0255	0.0040	0.0044	0.0145	0.0208	0.0061	0.0133
	β	–	–	–	–	0.0028	0.0009	0.6662	2.4672
CRT AEE	Alley1	0.1845	0.1826	0.5338	0.5645	0.4856	0.5213	0.1800	0.1562
	Ambush7	1.9943	1.8338	2.7624	3.0104	2.6527	2.9034	1.9578	1.5426
	Bamboo2	0.7449	0.7364	1.1076	1.1346	1.0569	1.0860	0.7379	0.6954
	Cave4	0.7764	0.7214	1.2970	1.3405	1.2070	1.2636	0.8041	0.6479
	Market2	0.8813	0.8847	1.4304	1.5059	1.3614	1.4237	0.8718	0.8333
	Market6	8.5738	8.8461	6.9282	7.6647	6.8884	7.6072	7.2835	7.2343
	Shaman2	0.1583	0.1554	0.4923	0.5036	0.4486	0.4628	0.1544	0.1331
	Sleeping1	0.0957	0.1002	0.3776	0.3668	0.3370	0.3315	0.0810	0.0616
	Temple3	15.434	17.661	20.035	20.143	19.577	19.222	15.684	17.224
CRT	**AVG AEE**	3.2048	3.4580	3.8849	4.0259	3.7794	3.8691	3.0838	3.1698
	α	0.0065	0.0096	0.0008	0.0006	0.0205	0.0376	0.0070	0.0080
	β	–	–	–	–	0.0008	0.0006	2.2779	5.7065

Table 5.7: Complete set of error data with corresponding weights achieved during evaluation of the Sintel data set.

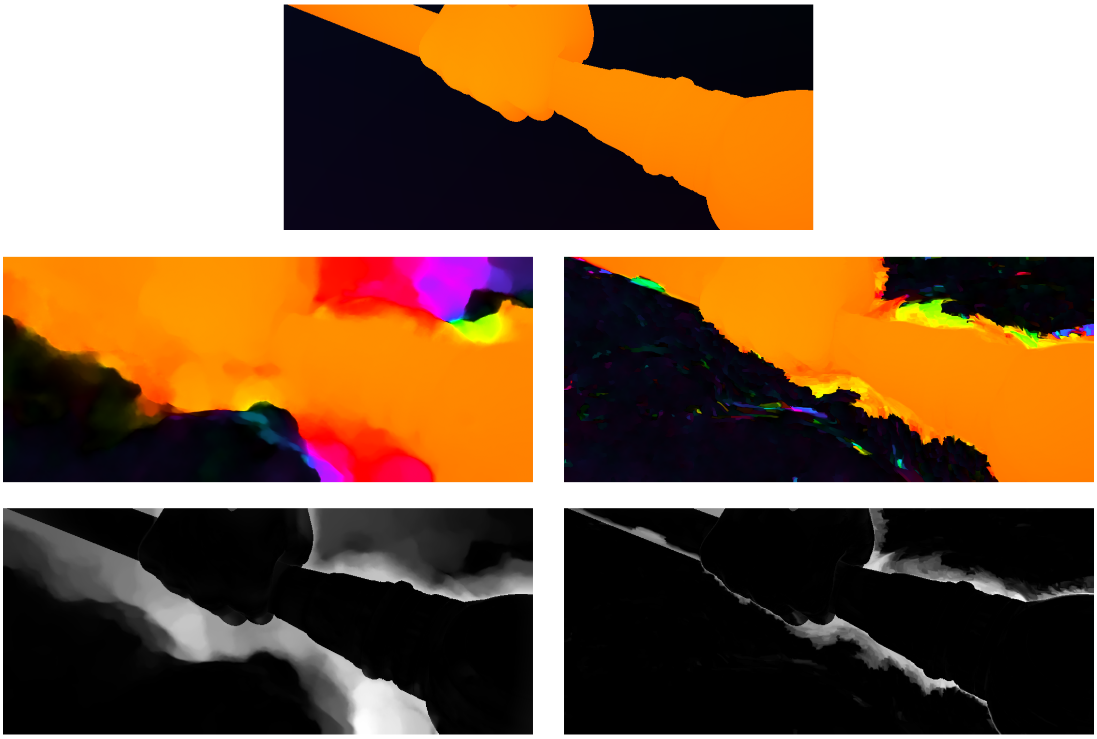

Figure 5.10: Comparison between the BCA and the CRT using the Ambush7 sequence. **First row:** Ground truth. **Left to Right:** BCA first order isotropic and CRT coupled anisotropic regularization. **Top to Bottom:** Motion field and difference image.

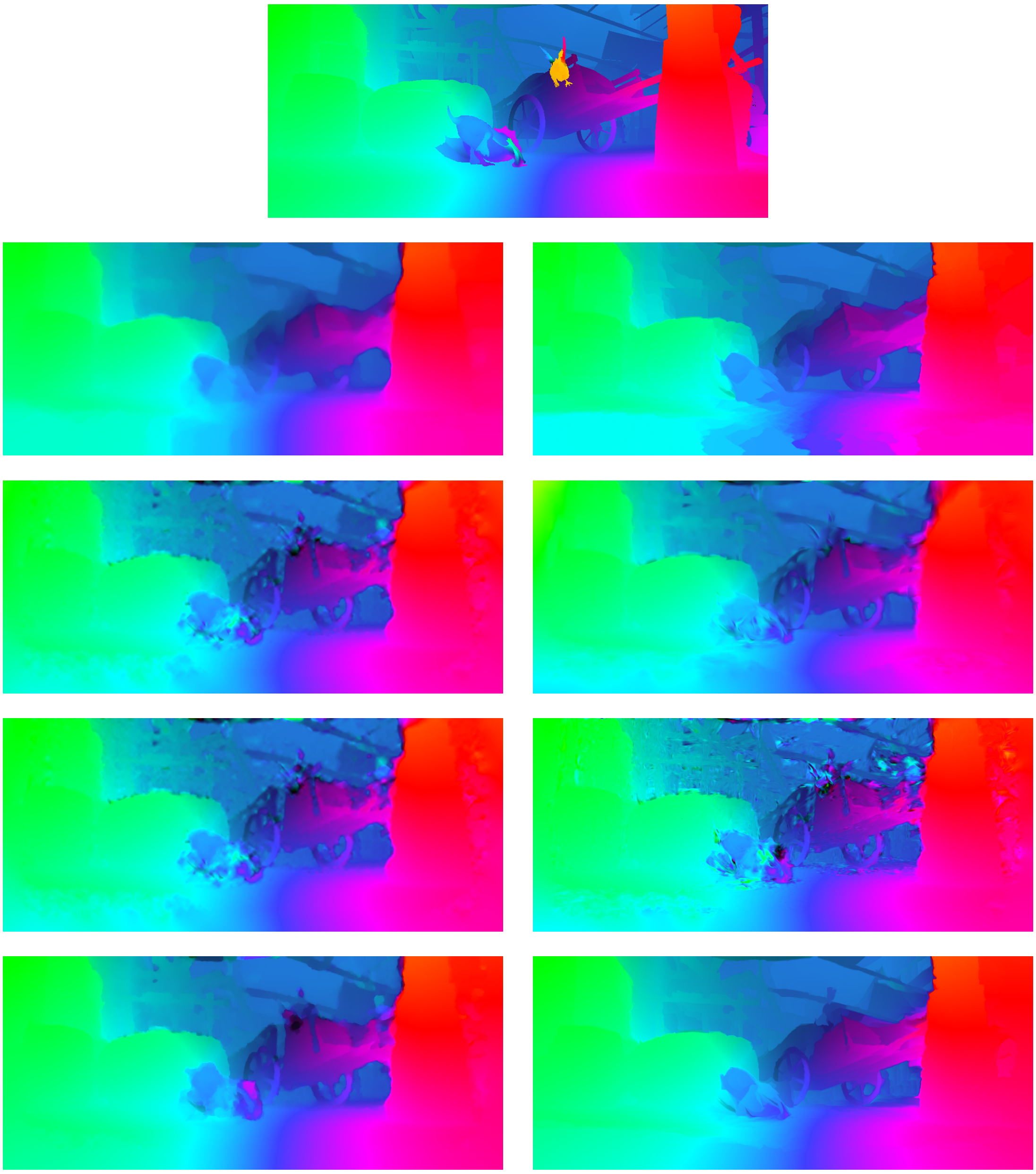

Figure 5.11: Results for the Market6 sequence achieved using the GCA. **First row:** Ground truth. **Left to Right:** Isotropic and anisotropic regularization. **Top to Bottom:** First order, second order, combined and coupled regularization.

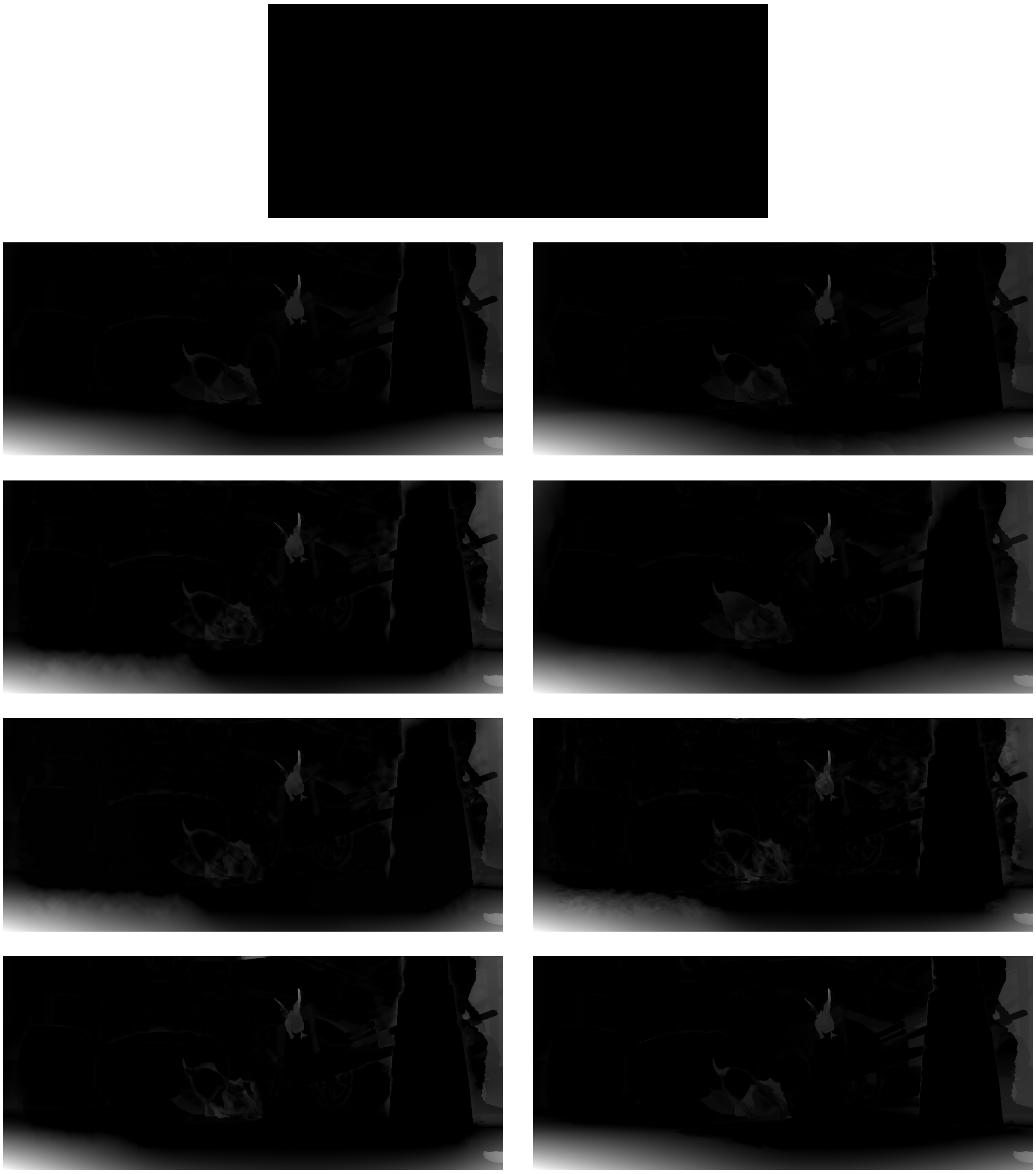

Figure 5.12: Differences to ground truth of corresponding regularizers using the GCA. **First row:** No difference. **Left to Right:** Isotropic and anisotropic regularization. **Top to Bottom:** First order, second order, combined and coupled regularization.

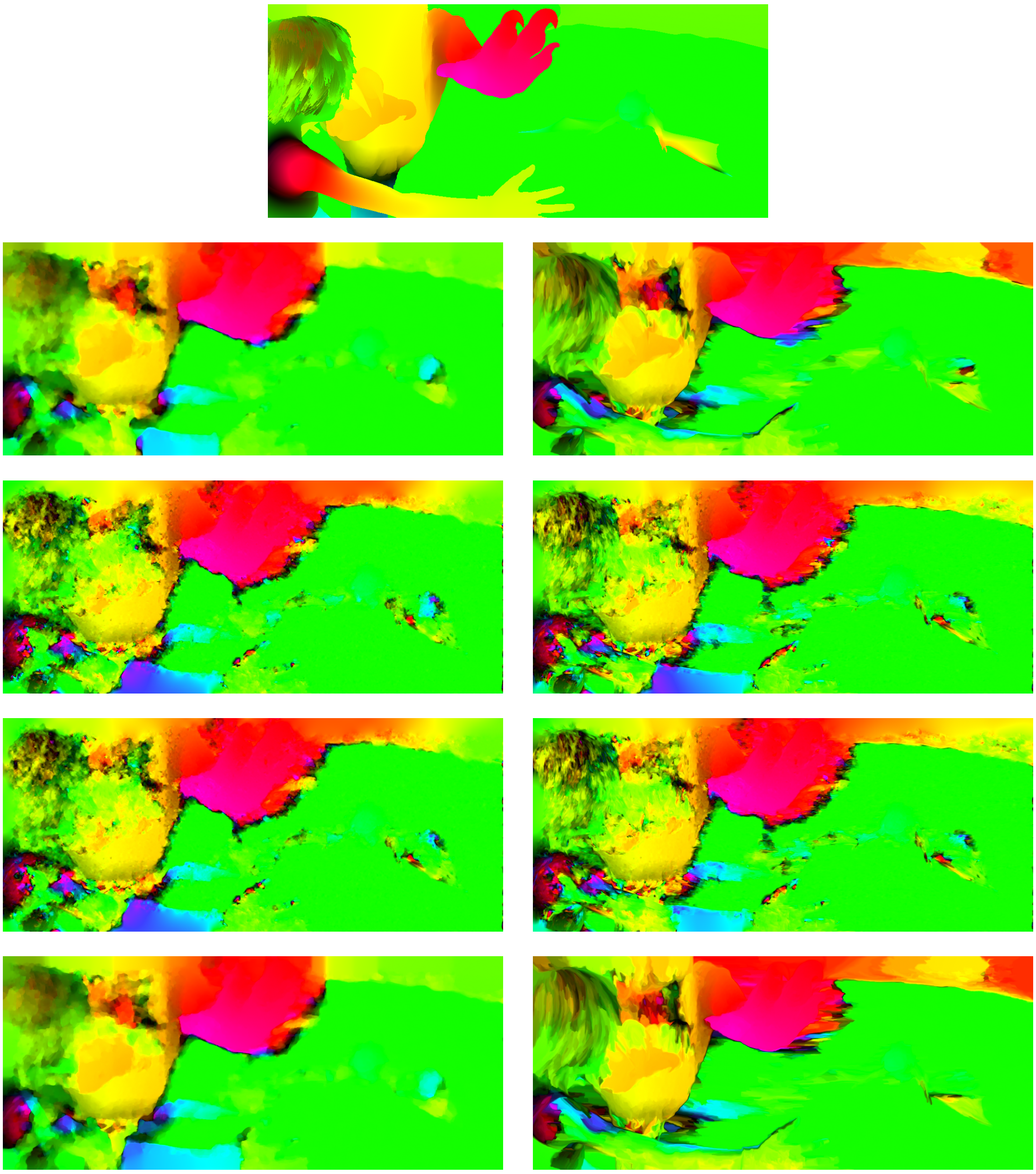

Figure 5.13: Results for the Temple3 sequence achieved using the CRT. **First row:** Ground truth. **Left to Right:** Isotropic and anisotropic regularization. **Top to Bottom:** First order, second order, combined and coupled regularization.

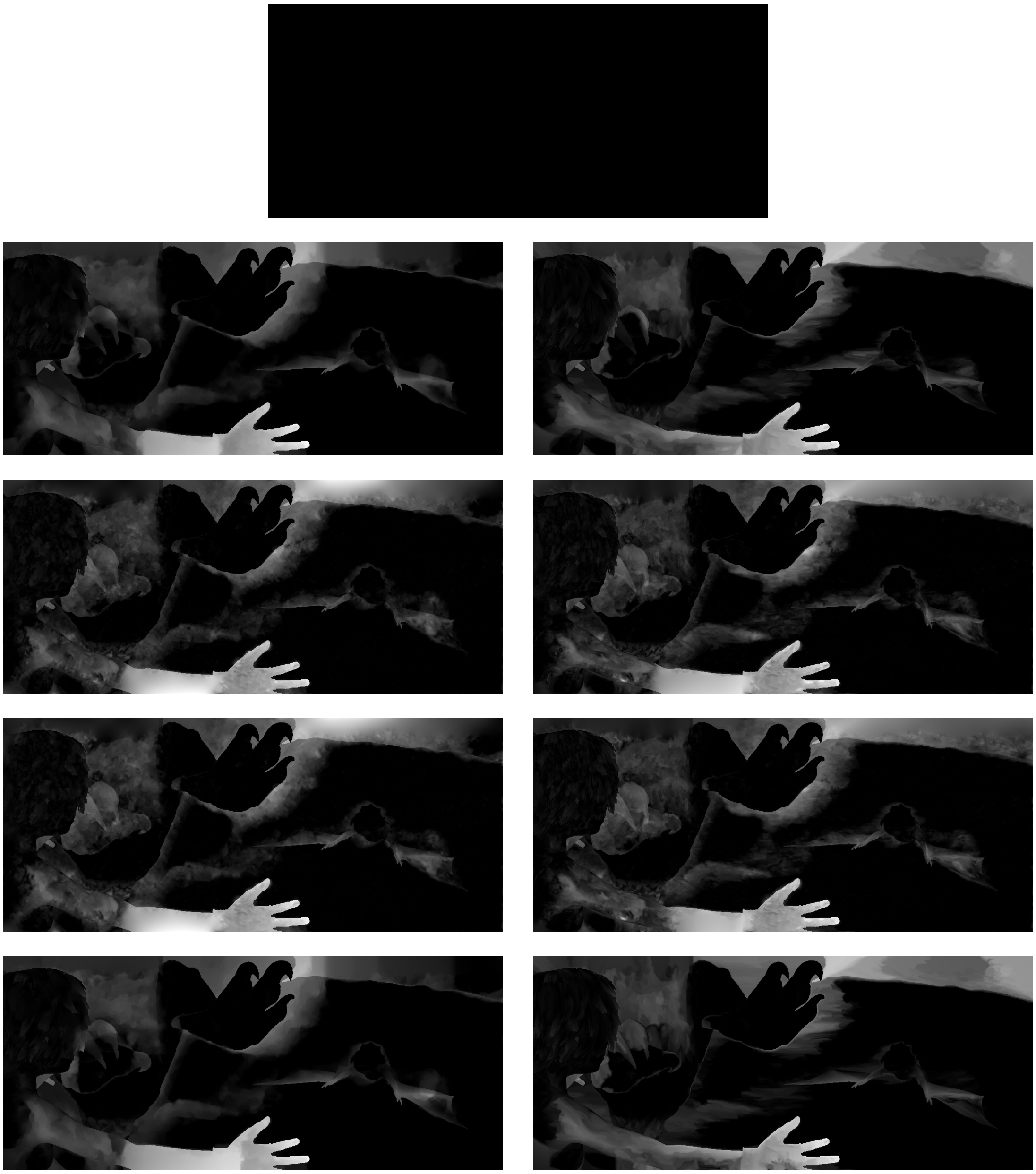

Figure 5.14: Differences to ground truth of corresponding regularizers using the CRT. **First row:** No difference. **Left to Right:** Isotropic and anisotropic regularization. **Top to Bottom:** First order, second order, combined and coupled regularization.

5.2.3 Results - KITTI

The KITTI data set contains test images taken in the real world that are now used in our last evaluation. These images are captured in a realistic application scenario, namely autonomous driving. Thus, they make the most naturalistic demands to optical flow estimation. To show the impact of the constancy assumptions, sequence 29 is a good example. One half of this sequence is illuminated by bright sunlight, the other half is covered in dark shadows. We choose the regularizer that gives the best result for each constancy assumption based on the error listing in Table 5.8 and compare them in Figure 5.15. One first observation that we make is that in case of the GCA and the CRT best results are achieved with combined regularization. The advantage of regularizers incorporating second order derivatives is not surprising since a camera mounted on a driving car is nothing else than egomotion of the camera and hence affine solutions are superior to first order regularization. Using the CRT for sequence 29, the error of first order regularization is even reduced by a factor of more than 2 in the combined case. The visualization also shows that CRT combined regularization creates some artifacts in the upper half of the image, but they are not covered by the error measurement since KITTI ground truth images only have a density of approximately 50%.

Let us now discuss the average results given by Table 5.8 in detail: (1) Employing the same regularizers, the GCA always achieves lower errors than the BCA, and the CRT further improves the results of the GCA (except first order regularization). (2) Anisotropic regularization decreases the error in 7 out of 12 cases. (3) While the best results per constancy assumption are always obtained with coupled regularization, the worst results are inconsistent as the red colored values indicate. (4) Second order and combined regularization work better with the CRT applied.

Results for the individual sequences further reveal, that the CRT not only yields the best average value with most of the regularizers, it is also less dependent on anisotropic regularization. In fact, the best AEE is produced by CRT coupled isotropic regularization.
But let us look on sequence 46 now. The capturing camera stands still and only one other vehicle is moving through the scenery (translational motion). BCA and GCA first order as well as coupled regularization need their anisotropic variant to produce good results here (orange errors). This is not the case with the CRT. Even CRT second order and combined regularization achieve relatively good results in this sequence. Figure 5.16, Figure 5.17, Figure 5.18 and Figure 5.19 show a comparison between regularization with the GCA or the CRT applied for sequence 46. While the CRT produces more artifacts in the sky compared to the GCA, the motion of the car is visible for all regularizers. Nevertheless, the best result is achieved by GCA coupled anisotropic regularization.
To show an example where anisotropic regularization always deteriorates the estimation (compared to its isotropic counterpart), we visualize results for sequence 120 in Figure 5.20 and Figure 5.21. Optical flow estimation is based on the GCA here. Similarly to the results of sequence 46 second order and combined regularization suffer from artifacts in the bright sky, while first order and coupled regularization produce more stable results. Again, these artifacts are not counted towards the error, since the ground truth does not provide any data in this region.

KITTI		First Order		Second Order		Combined Regularization		Coupled Regularization	
		Iso	Aniso	Iso	Aniso	Iso	Aniso	Iso	Aniso
BCA AEE	14	0.9226	0.8970	1.4538	2.5870	1.7582	2.6027	0.8589	0.8459
	29	24.683	25.062	23.706	28.204	26.953	28.384	23.610	23.482
	46	10.337	2.3902	14.761	13.513	12.716	13.773	11.203	2.4722
	86	5.7883	3.9429	11.781	7.2031	7.0265	7.1969	4.6096	3.9909
	120	16.949	15.943	11.209	13.431	11.461	13.454	12.168	11.838
	144	1.5696	1.9146	1.7597	3.7090	2.6504	3.7534	1.0975	1.1182
	171	3.1978	2.7850	3.2434	4.7188	3.6722	4.7343	2.0374	2.2820
BCA	**AVG AEE**	9.0638	7.5622	9.7020	10.481	9.4625	10.557	7.9407	6.5755
	α	0.0046	0.0078	0.0021	0.0003	0.0437	0.0294	0.0046	0.0077
	β	–	–	–	–	0.0006	0.0002	0.9210	0.9115
GCA AEE	14	0.8862	0.9738	1.7789	2.1004	1.7135	2.0511	1.1297	0.8315
	29	17.704	16.753	14.731	15.413	13.243	15.483	14.297	15.002
	46	7.8856	2.7075	17.234	17.459	17.151	17.257	17.041	2.3396
	86	7.6118	5.8279	7.2360	7.5679	7.1564	6.8249	5.3009	4.7692
	120	11.418	12.124	6.1870	9.3711	6.2641	9.2110	6.6339	9.1047
	144	0.8633	0.9651	1.6534	1.9186	1.5834	1.8635	0.5001	0.5359
	171	1.9194	2.0291	2.1879	2.6202	2.1150	2.4427	1.0972	1.2598
GCA	**AVG AEE**	6.8982	5.9114	7.2869	8.0643	7.0323	7.8762	6.5714	4.8346
	α	0.0047	0.0071	0.0008	0.0005	0.0240	0.0133	0.0102	0.0082
	β	–	–	–	–	0.0006	0.0004	1.0864	1.0707
CRT AEE	14	1.8566	1.5337	2.5683	2.6056	2.4147	2.5710	1.0342	1.0326
	29	14.257	17.324	6.7378	6.6676	6.5256	6.7697	8.3150	8.7290
	46	6.0500	4.4396	5.8605	5.4616	5.6998	5.3263	3.8447	4.1267
	86	9.9069	6.5300	10.011	10.111	9.9541	9.9331	7.1974	5.1656
	120	12.451	11.934	8.1360	8.3602	7.9692	8.2484	7.3349	8.7914
	144	4.2597	3.5567	6.9041	6.1198	6.5938	5.9449	1.8827	2.6168
	171	4.1051	2.9994	3.7772	3.6762	3.5003	3.6160	1.8967	1.7876
CRT	**AVG AEE**	7.5551	6.9025	6.2850	6.1431	6.0939	6.0585	4.5008	4.6071
	α	0.0010	0.0024	0.0001	0.0001	0.0554	0.0135	0.0036	0.0044
	β	–	–	–	–	0.0001	0.0001	0.3581	0.4199

Table 5.8: Complete set of error data with corresponding weights achieved during evaluation of the KITTI data set.

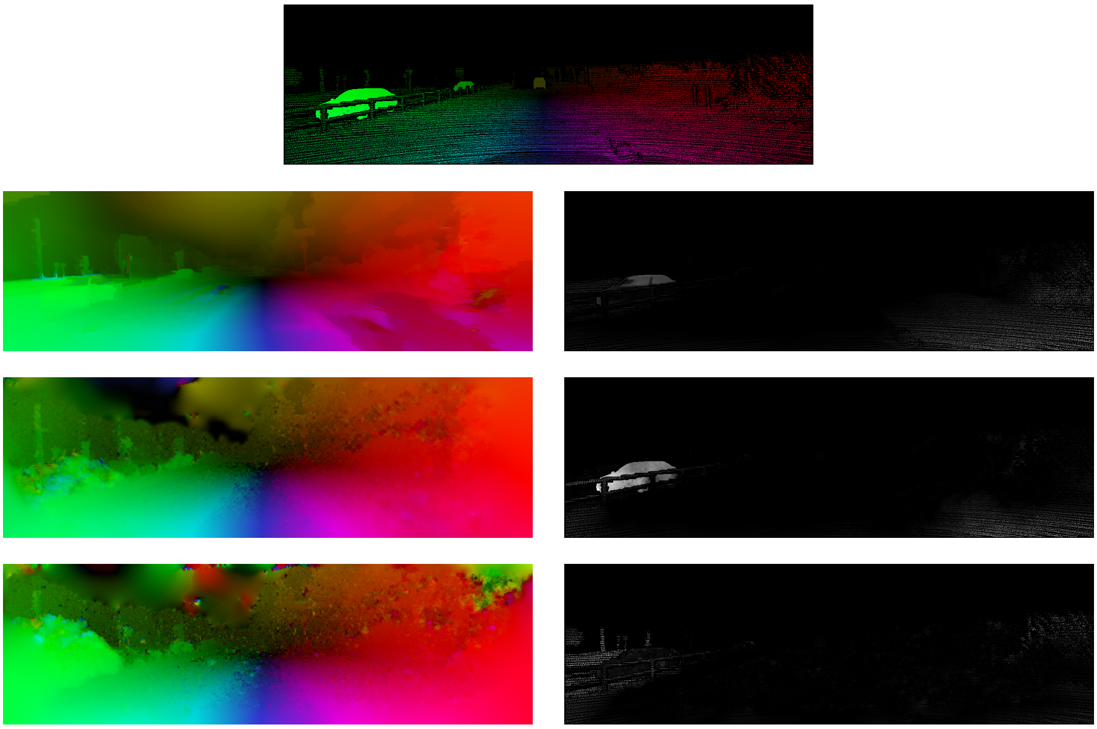

Figure 5.15: Comparison between the BCA, the GCA and the CRT using sequence 29. **First row:** Ground truth. **Left to Right:** Motion field and difference image. **Top to Bottom:** BCA coupled anisotropic, GCA combined isotropic and CRT combined isotropic regularization.

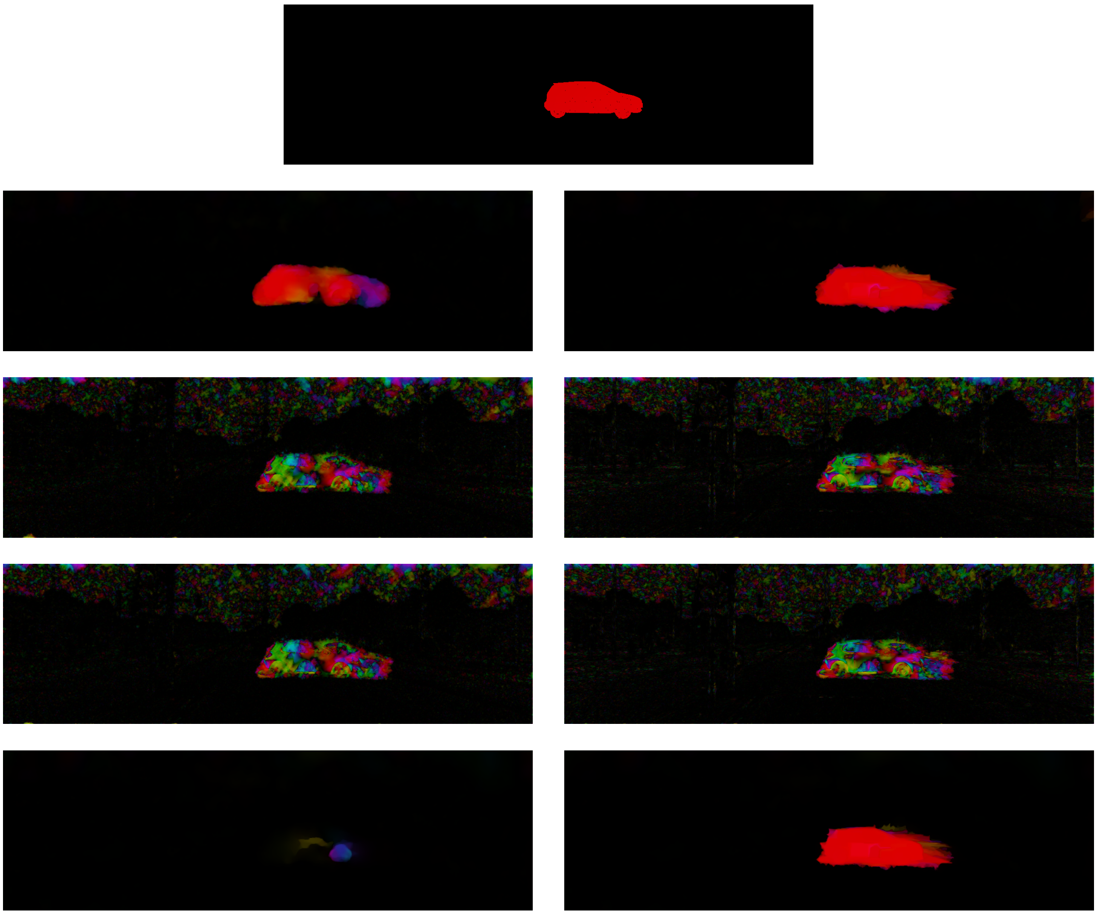

Figure 5.16: Results for sequence 46 using different regularizers and the GCA. **First row:** Ground truth. **Left to Right:** Isotropic and anisotropic regularization. **Top to Bottom:** First order, second order, combined and coupled regularization.

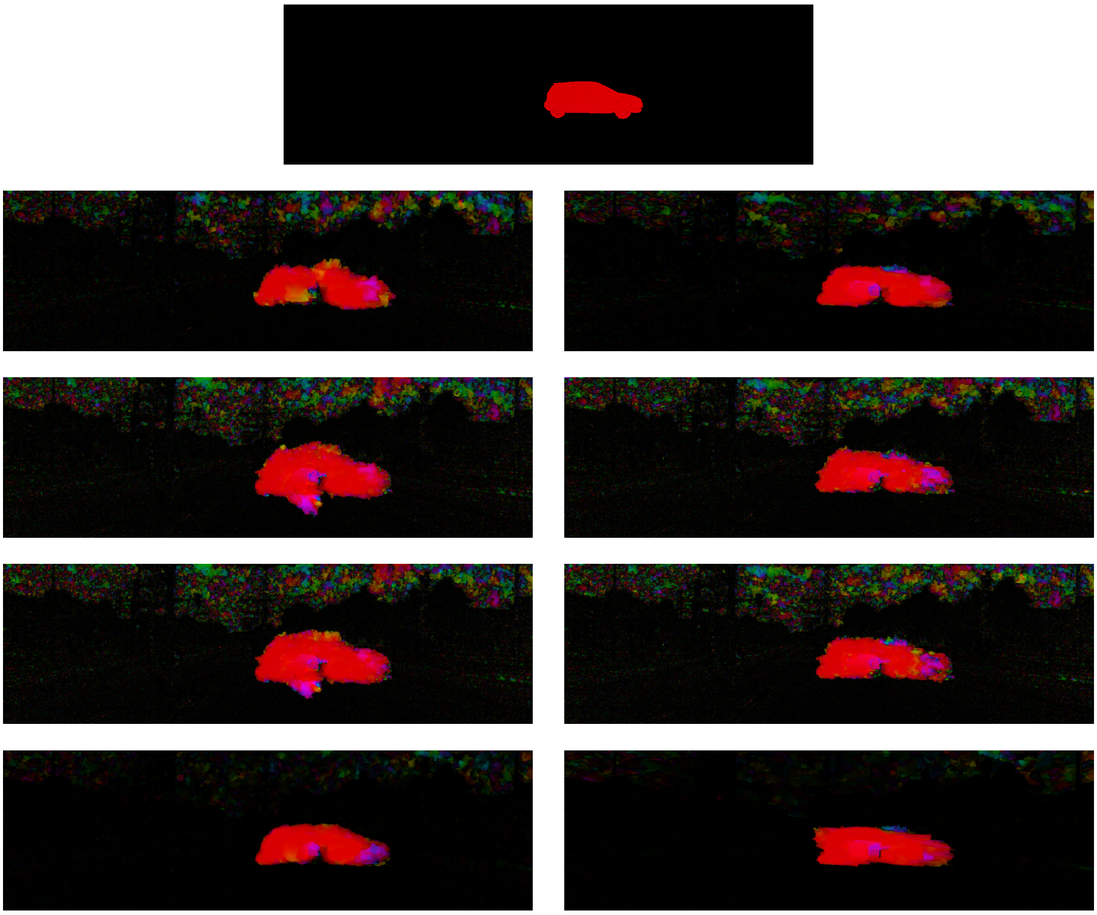

Figure 5.17: Results for sequence 46 using different regularizers and the CRT. **First row:** Ground truth. **Left to Right:** Isotropic and anisotropic regularization. **Top to Bottom:** First order, second order, combined and coupled regularization.

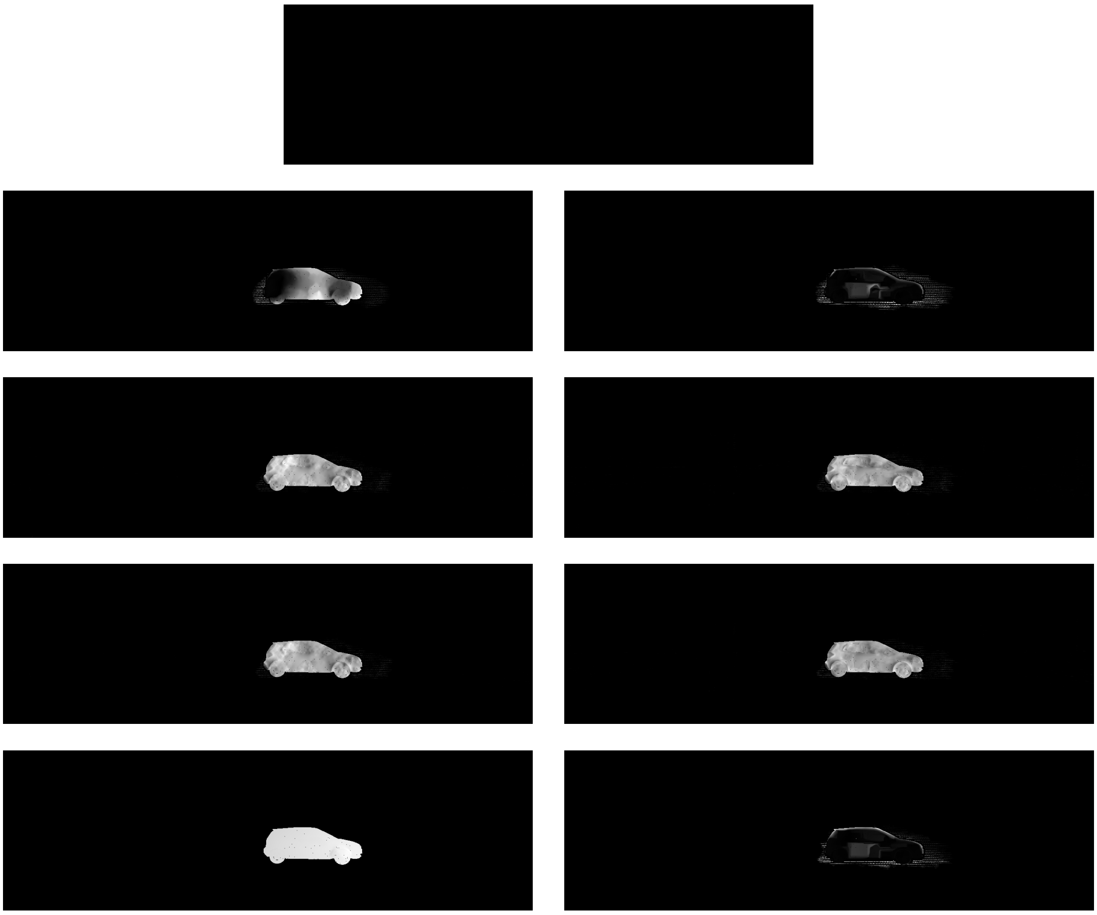

Figure 5.18: Differences to ground truth of corresponding regularizers using the GCA. **First row:** No difference. **Left to Right:** Isotropic and anisotropic regularization. **Top to Bottom:** First order, second order, combined and coupled regularization.

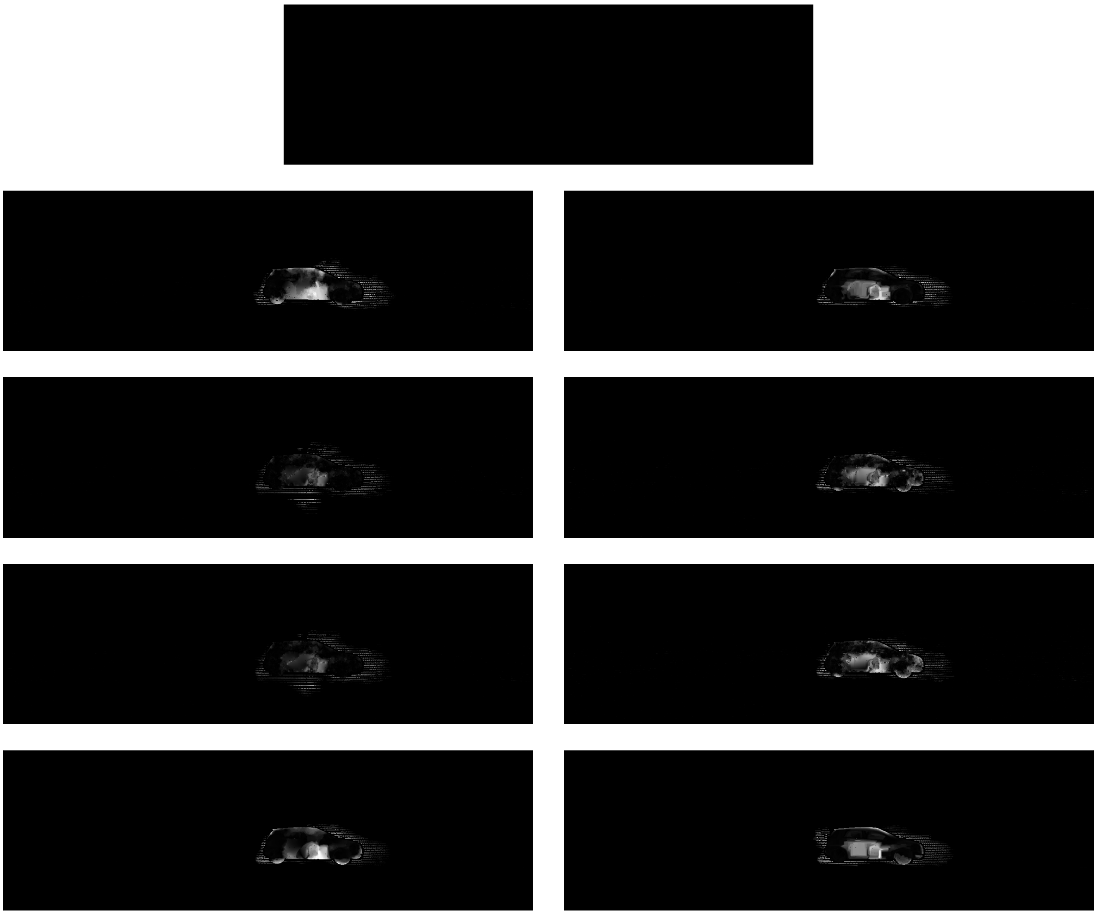

Figure 5.19: Differences to ground truth of corresponding regularizers using the CRT. **First row:** No difference. **Left to Right:** Isotropic and anisotropic regularization. **Top to Bottom:** First order, second order, combined and coupled regularization.

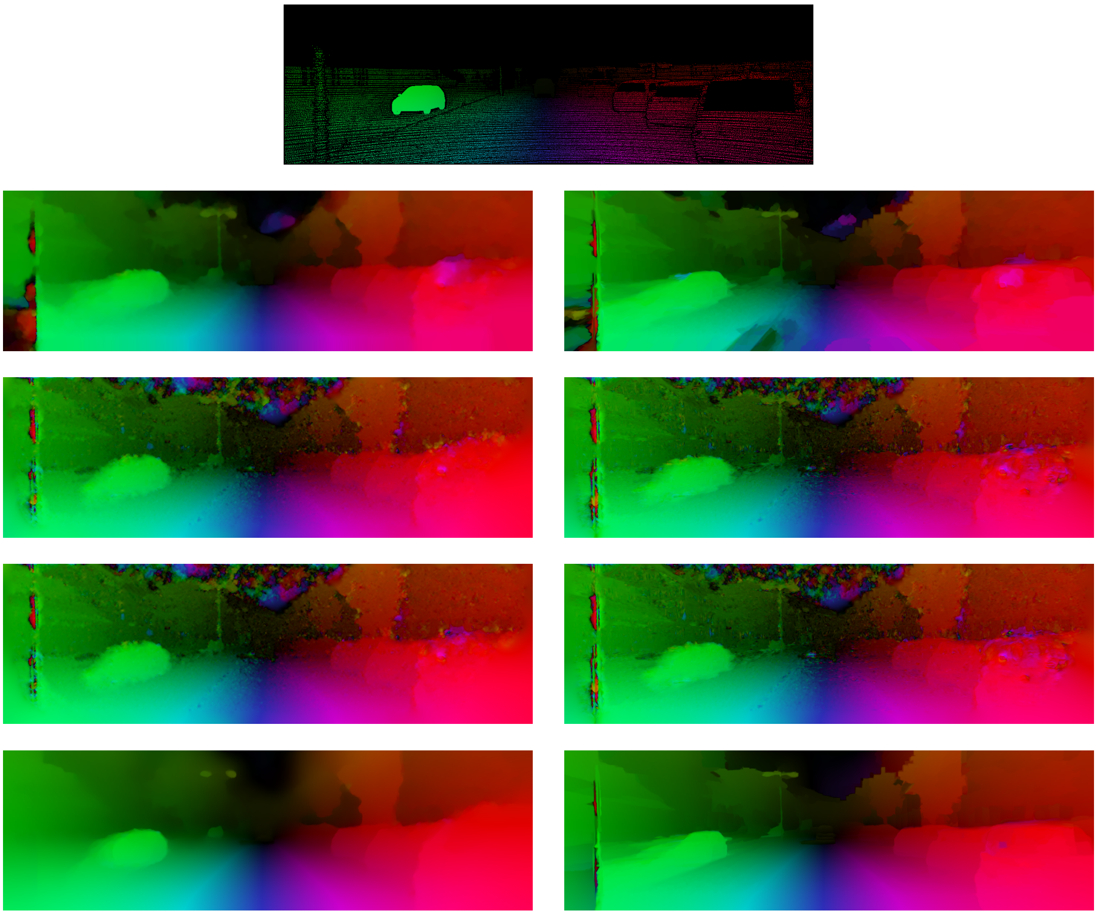

Figure 5.20: Results for sequence 120 using different regularizers and the GCA. **First row:** Ground truth. **Left to Right:** Isotropic and anisotropic regularization. **Top to Bottom:** First order, second order, combined and coupled regularization.

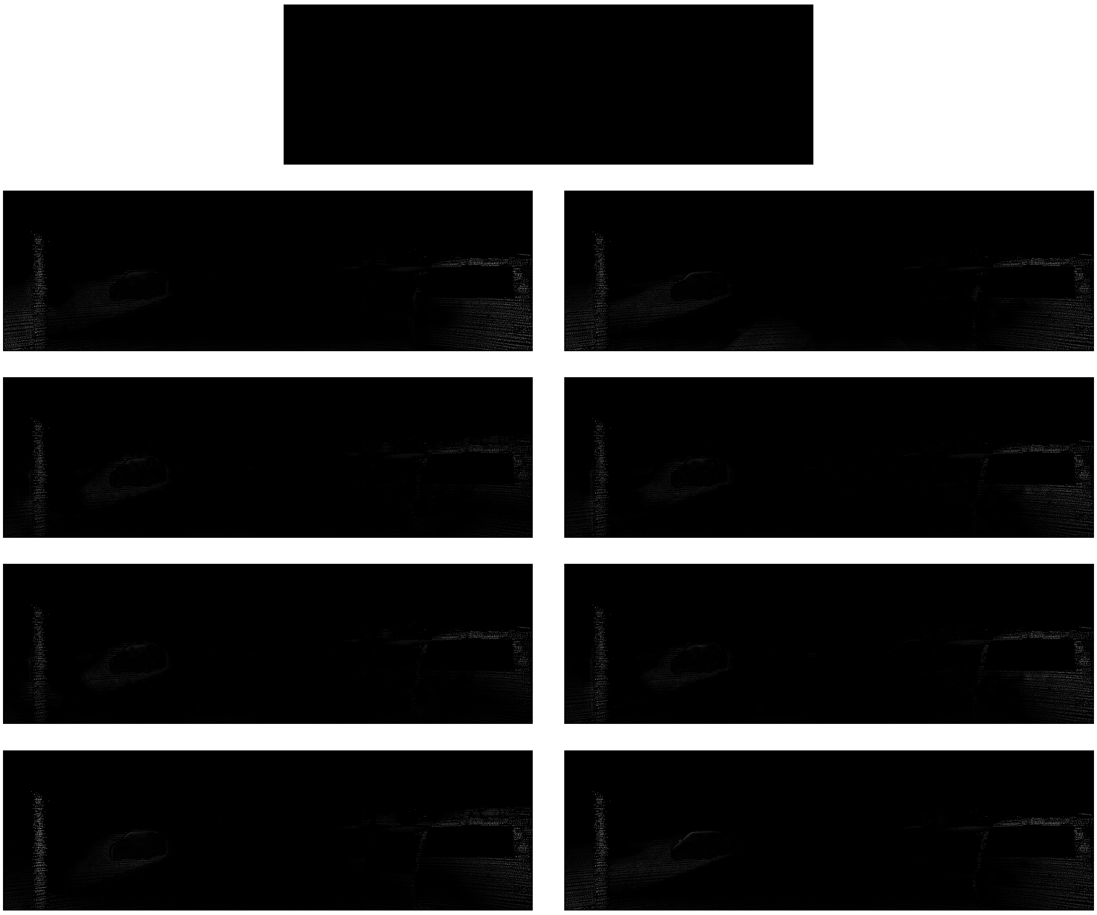

Figure 5.21: Differences to ground truth of corresponding regularizers using the GCA. **First row:** No difference. **Left to Right:** Isotropic and anisotropic regularization. **Top to Bottom:** First order, second order, combined and coupled regularization.

5.3 Summary

Because the eight different regularizers have been evaluated with all the selected test images from the data sets at this point, we will now summarize some final insights gathered through our results.

Let us first focus on the different constancy assumptions with respective AEEs listed in Table 5.6, Table 5.7 and Table 5.8. With the Middlebury test images still producing competing estimations, the BCA was not capable to provide good results under complex lighting conditions as certain sequences from the Sintel or the KITTI data set showed. In average the CRT achieved more accurate and stable results, but in individual cases the GCA even outperformed the CRT. It is quite difficult to choose the GCA or the CRT for a certain scene. If someone, on the one hand, is interested in the best average performance possible, we recommend using the CRT, but if computation speed plays a major role on the other hand, we suggest the GCA as a good compromise (runtime overview given in Table 5.9).

We now discuss the results of the different regularizers in detail.
First order regularization is yet a simple, but still very effective technique to regularize an unknown flow field. Using the Middlebury or the Sintel test images its average errors were very close to best results for each constancy assumption, if not even the best result achieved in total (Middlebury GCA). But the Market6 sequence of the Sintel data set already hinted at a drawback of using first order derivatives. Scenes with strong affine motion caused by fast egomotion of the camera make first order regularization yield worse results than usual. This behavior was confirmed during evaluation of the KITTI data set (sequence 29 and 120). Regarding the difference between first order isotropic and anisotropic regularization, only sequences from the KITTI data set showed a noticeable improvement of estimation quality with direction-dependent smoothing. The results of the other data sets were around the same level or even worsened by anisotropic regularization.
While first order regularization gave bad results in scenes with strong affine motion, second order regularization actually benefited only from these particular scenes. In fact, the worst results of the Middlebury data set with almost no affine motion present were given by second order regularization (except the GCA case, where combined regularization produced virtually the same results). The same behavior can be observed for the Sintel data set. There even the anisotropic approach achieved the worst average AEEs. Only during the last evaluation based on the KITTI data set this changed a little. With the CRT applied second order regularization achieved a lower average error than first order regularization for the first time. We expect second order regularization to give even better results if the selected data set contained more sequences with a driving car since under such circumstances affine motion is present. In the realistic scenario of autonomous driving a moving vehicle should usually be the norm. A real advantage of second order anisotropic over isotropic regularization cannot be determined. In many cases the anisotropic variant even gave worse results.
Observations for combined regularization were quite unexpected. This technique reaches very similar results as its second order isotropic and anisotropic counterparts with slightly lower average AEEs for combined regularization in most of the cases (except Middlebury GCA and KITTI BCA). Here, the following rule applies: If second order anisotropic regularization yields a better AEE in average than its isotropic variant, combined anisotropic

regularization also gives a better AEE in average than combined isotropic regularization with the combined AEE being lower than the second order AEE and the other way around as well (isotropic better than anisotropic). But first order regularization usually gave better results yet - except in the case of the special sequences exhibiting affine motion. Visualizations also reveal very similar results between second order and combined regularization.
The best average results per constancy assumption were achieved with coupled regularization almost every time. Only the Middlebury and Sintel images using the GCA favored first order regularization. Coupled regularization can handle scenes that contain only constant motion, but it is also capable to produce good results with affine motion. In single cases first order, second order or combined regularization can yield a more accurate estimation, but coupled regularization wins in average. Together with the CRT it achieves the best total results in its isotropic variant for the Sintel and KITTI test images. Again, a clear advantage of anisotropic regularization over isotropic regularization cannot be demonstrated.

Runtime. So far we only concentrated on quality of estimation in terms of error metrics and optimized parameters to reduce the difference to a ground truth image. But even the best technique is affected if the result takes too long to be computed. Hence, we end this chapter with some final words on the runtime of our regularizers.
To show the influence of image size we compare the Venus sequence having the smallest images (159600 pixel) with sequence 120 since the KITTI data set provides the largest images (465750 pixel). This corresponds to an increase of pixels by a factor of approximately 3. The time that is necessary to compute the optical flow fields is listed in Table 5.9. We can see that anisotropic regularization takes longer as its isotropic counterpart in each case. While the difference might be unrecognizable first, it increases with each following regularization technique. First and second order regularization are roughly at the same level, followed by combined regularization and coupled regularization takes the most time. Time is especially then increased if the CRT is applied. This is not surprising since we need to compute a neighborhood with many constraints for each pixel here. Between the Venus sequence and sequence 120 the necessary amount of time is also roughly increased by a factor of 3.

Runtime		**First Order**		**Second Order**		**Combined Regularization**		**Coupled Regularization**	
		Iso	Aniso	Iso	Aniso	Iso	Aniso	Iso	Aniso
Venus	BCA	4.1	4.7	4.6	6.5	7.8	9.8	11.0	17.9
	GCA	4.2	4.7	4.8	6.8	7.9	10.1	11.6	18.3
	CRT	13.1	13.5	13.7	15.0	16.5	18.4	20.5	27.4
120	BCA	10.9	12.3	12.1	16.7	20.0	26.1	29.8	52.2
	GCA	12.1	13.0	13.4	17.9	21.8	27.1	31.2	53.3
	CRT	44.4	45.6	45.6	49.7	54.8	60.3	65.3	87.1

Table 5.9: Runtime in seconds for each regularizer on two different image sizes (Venus sequence with 420 × 380 pixel and sequence 120 with 1242 × 375 pixel).

6 Conclusion

Regularizers of different order in an isotropic as well as an anisotropic setting for variational motion estimation have been the main topic in this thesis. We gave an introduction to data terms, proposed ideas to regularize an unknown solution, explained various extensions and brought all these things together in a revealing comparison.
Before we were able to discuss details of optical flow estimation, some foundations and basic notations were clarified in Chapter 2.
Subsequently, an introduction to variational methods has been given in Chapter 3. We discussed the several parts of variational motion estimation, namely data term and regularization (smoothness) term, and their influence on the final result. With respect to the data term we started with different constancy assumptions and incorporated some modifications as color images, normalization and different penalizers to further increase accuracy. Also a suitable minimization strategy was derived to solve the problem numerically. Regarding the regularization term we only gave a brief introduction since an in-depth explanation followed in Chapter 4.
Derivatives of different order represent different kinds of motion. This was the main statement that our considerations based upon in this chapter. Is it not a good idea to connect the various derivatives in some way to take these motion types into account during regularization? Investigating the impact of the derivatives was of great interest and hence we first introduced regularization techniques with only first or second order derivatives and secondly came up with two different concepts to make the connection between them. Each of the initial isotropic regularizers was modified to enable direction-dependent smoothing resulting in sharper motion edges.
Finally, suitable data sets were needed to give a realistic overview of performance. Therefore, we chose three prominent data sets and selected individual test sequences from each of them. In Chapter 5, these sequences were presented - followed by the results of the evaluation.

The evaluation shows that regularization with first and second order derivatives is highly beneficial in a variational optical flow context. First order regularization can achieve good results in many test sequences already, but especially the realistic KITTI data set with numerous scenes containing affine motion unfolds the limits of this technique. In these scenes second order regularization performs better, but fails to reach the errors of the first order one in other scenes in contrast. Combined regularization gives error rates rather similar to the second order method, but coupled regularization showed that connecting first and second order derivatives can actually yield the most accurate estimations. Regarding isotropic versus anisotropic regularization, a clear advantage of direction-dependent smoothing cannot be determined. Take the KITTI benchmark for example: Using the BCA or the GCA, coupled anisotropic regularization gave the best result. But the best result in total was achieved by the CRT coupled isotropic approach.

Outlook. Our proposed methods realizing regularization of different order still offer some room for improvements which are mentioned now.

- **Occlusions:** Experiments showed that scenes with large or affine motion often suffer from occlusions. In such areas correspondences cannot be found accurately which ultimately increases errors there. Therefore, occlusion handling would be reasonable.
- **Multi-frame optical flow:** So far only two images have been used to estimate motion fields. Since motion fields of consecutive image pairs are usually coherent, incorporating multiple frames and thus allowing to temporarily regularize multiple solutions along motion trajectories [46] can improve accuracy.
- **Separate robustification:** During the introduction of anisotropy to second order regularization we applied a joint robustification. To increase the degree of anisotropy even further second order derivatives or auxiliary vectors in the combined case can be penalized separately with different combinations of the eigenvectors extracted from the regularization tensor.

A Regularization Stencils

A.1 First Order Isotropic

	$i-1$	i	$i+1$
$j-1$		$\frac{1}{2h_y^2}(\Psi'_{i,j} + \Psi'_{i,j-1})$	
j	$\frac{1}{2h_x^2}(\Psi'_{i,j} + \Psi'_{i-1,j})$	$-\frac{1}{2h_x^2}(\Psi'_{i+1,j} + 2\Psi'_{i,j} + \Psi'_{i-1,j})$ $-\frac{1}{2h_y^2}(\Psi'_{i,j+1} + 2\Psi'_{i,j} + \Psi'_{i,j-1})$	$\frac{1}{2h_x^2}(\Psi'_{i,j} + \Psi'_{i+1,j})$
$j+1$		$\frac{1}{2h_y^2}(\Psi'_{i,j} + \Psi'_{i,j+1})$	

Table A.1: Stencil for a first order isotropic regularizer.

A.2 First Order Anisotropic

	$i-1$	i	$i+1$
$j-1$	$\frac{\lvert b_{i-1,j-1}\rvert + b_{i-1,j-1}}{4h_xh_y}$ $+\frac{\lvert b_{i,j}\rvert + b_{i,j}}{4h_xh_y}$	$\frac{c_{i,j-1}+c_{i,j}}{2h_y^2} - \frac{\lvert b_{i,j-1}\rvert + \lvert b_{i,j}\rvert}{2h_xh_y}$	$\frac{\lvert b_{i+1,j-1}\rvert - b_{i+1,j-1}}{4h_xh_y}$ $+\frac{\lvert b_{i,j}\rvert - b_{i,j}}{4h_xh_y}$
j	$\frac{a_{i-1,j}+a_{i,j}}{2h_x^2}$ $-\frac{\lvert b_{i-1,j}\rvert + b_{i,j}}{2h_xh_y}$	$-\frac{a_{i-1,j}+2a_{i,j}+a_{i+1,j}}{2h_x^2}$ $-\frac{\lvert b_{i-1,j+1}\rvert - b_{i-1,j+1} + \lvert b_{i+1,j+1}\rvert + b_{i+1,j+1}}{4h_xh_y}$ $-\frac{\lvert b_{i-1,j-1}\rvert + b_{i-1,j-1} + \lvert b_{i+1,j-1}\rvert - b_{i+1,j-1}}{4h_xh_y}$ $+\frac{\lvert b_{i-1,j}\rvert + \lvert b_{i+1,j}\rvert + \lvert b_{i,j-1}\rvert + \lvert b_{i,j+1}\rvert + 2\lvert b_{i,j}\rvert}{2h_xh_y}$ $-\frac{c_{i,j-1}+2c_{i,j}+c_{i,j+1}}{2h_y^2}$	$\frac{a_{i+1,j}+a_{i,j}}{2h_x^2}$ $-\frac{\lvert b_{i+1,j}\rvert + b_{i,j}}{2h_xh_y}$
$j+1$	$\frac{\lvert b_{i-1,j+1}\rvert - b_{i-1,j+1}}{4h_xh_y}$ $+\frac{\lvert b_{i,j}\rvert - b_{i,j}}{4h_xh_y}$	$\frac{c_{i,j+1}+c_{i,j}}{2h_y^2} - \frac{\lvert b_{i,j+1}\rvert + \lvert b_{i,j}\rvert}{2h_xh_y}$	$\frac{\lvert b_{i+1,j+1}\rvert + b_{i+1,j+1}}{4h_xh_y}$ $+\frac{\lvert b_{i,j}\rvert + b_{i,j}}{4h_xh_y}$

Table A.2: Nonnegative stencil [42] for a first order anisotropic regularizer.

A.3 Second Order Isotropic

	$i-2$	$i-1$	i	$i+1$	$i+2$
$j-2$	$\frac{1}{8h_x^2h_y^2}\left(\Psi'_{i-1,j-1}\right)$		$-\frac{1}{8h_x^2h_y^2}\left(\Psi'_{i+1,j-1}+\Psi'_{i-1,j-1}\right)$ $+\frac{1}{h_y^4}\left(\Psi_{i,j-1}\right)$		$\frac{1}{8h_x^2h_y^2}\left(\Psi'_{i+1,j-1}\right)$
$j-1$			$-\frac{2}{h_y^4}\left(\Psi'_{i,j}+\Psi'_{i,j-1}\right)$		
j	$\frac{1}{h_x^4}\left(\Psi_{i-1,j}\right)$ $-\frac{1}{8h_x^2h_y^2}\left(\Psi'_{i-1,j+1}+\Psi'_{i-1,j-1}\right)$	$-\frac{2}{h_x^4}\left(\Psi'_{i,j}+\Psi'_{i-1,j}\right)$	$\frac{1}{h_x^4}\left(\Psi'_{i+1,j}+4\Psi'_{i,j}+\Psi'_{i-1,j}\right)$ $+\frac{1}{8h_x^2h_y^2}\left(\Psi'_{i+1,j+1}+\Psi'_{i+1,j-1}\right.$ $\left.+\Psi'_{i-1,j+1}+\Psi'_{i-1,j-1}\right)$ $+\frac{1}{h_y^4}\left(\Psi'_{i,j+1}+4\Psi'_{i,j}+\Psi'_{i,j-1}\right)$	$-\frac{2}{h_x^4}\left(\Psi'_{i+1,j}+\Psi'_{i,j}\right)$	$\frac{1}{h_x^4}\left(\Psi_{i+1,j}\right)$ $-\frac{1}{8h_x^2h_y^2}\left(\Psi'_{i+1,j+1}+\Psi'_{i+1,j-1}\right)$
$j+1$			$-\frac{2}{h_y^4}\left(\Psi'_{i,j+1}+\Psi'_{i,j}\right)$		
$j+2$	$\frac{1}{8h_x^2h_y^2}\left(\Psi'_{i-1,j+1}\right)$		$-\frac{1}{8h_x^2h_y^2}\left(\Psi'_{i+1,j+1}+\Psi'_{i-1,j+1}\right)$ $+\frac{1}{h_y^4}\left(\Psi_{i,j+1}\right)$		$\frac{1}{8h_x^2h_y^2}\left(\Psi'_{i+1,j+1}\right)$

Table A.3: Stencil for a second order isotropic regularizer.

A.4 Second Order Anisotropic

	$i-2$	$i-1$	i	$i+1$	$i+2$
$j-2$	$\frac{1}{16h_x^2h_y^2}$ $(c_{i-1,j-1}+a_{i-1,j-1})$	$\frac{1}{4h_xh_y^3}$ $(b_{i-1,j-1}+b_{i,j-1})$	$-\frac{1}{16h_x^2h_y^2}(c_{i+1,j-1}+c_{i-1,j-1}$ $+a_{i+1,j-1}+a_{i-1,j-1})$ $+\frac{1}{h_y^4}(c_{i,j-1})$	$-\frac{1}{4h_xh_y^3}$ $(b_{i+1,j-1}+b_{i,j-1})$	$\frac{1}{16h_x^2h_y^2}$ $(c_{i+1,j-1}+a_{i+1,j-1})$
$j-1$	$\frac{1}{4h_x^3h_y}$ $(b_{i-1,j}+b_{i-1,j-1})$	$-\frac{1}{2h_x^3h_y}(b_{i,j}+b_{i-1,j-1})$ $-\frac{1}{2h_xh_y^3}(b_{i-1,j-1}+b_{i,j})$	$\frac{1}{4h_x^3h_y}(b_{i+1,j}-b_{i-1,j}$ $-b_{i+1,j-1}+b_{i-1,j-1})$ $-\frac{2}{h_y^4}(c_{i,j}+c_{i,j-1})$	$\frac{1}{2h_x^3h_y}(b_{i,j}+b_{i+1,j-1})$ $+\frac{1}{2h_xh_y^3}(b_{i+1,j-1}+b_{i,j})$	$-\frac{1}{4h_x^3h_y}$ $(b_{i+1,j}+b_{i+1,j-1})$
j	$\frac{1}{h_x^4}(a_{i-1,j})$ $-\frac{1}{16h_x^2h_y^2}(c_{i-1,j+1}+c_{i-1,j-1}$ $+a_{i-1,j+1}+a_{i-1,j-1})$	$-\frac{2}{h_x^4}(a_{i,j}+a_{i-1,j})$ $+\frac{1}{4h_xh_y^3}(b_{i-1,j-1}-b_{i-1,j+1}$ $+b_{i,j+1}-b_{i,j-1})$	$\frac{1}{h_x^4}(a_{i+1,j}+4a_{i,j}+a_{i-1,j})$ $+\frac{1}{16h_x^2h_y^2}(c_{i+1,j+1}+c_{i+1,j-1}$ $+c_{i-1,j+1}+c_{i-1,j-1}$ $+a_{i+1,j+1}+a_{i+1,j-1}$ $+a_{i-1,j+1}+a_{i-1,j-1})$ $+\frac{1}{h_y^4}(c_{i,j+1}+4c_{i,j}+c_{i,j-1})$	$-\frac{2}{h_x^4}(a_{i+1,j}+a_{i,j})$ $+\frac{1}{4h_xh_y^3}(b_{i+1,j+1}-b_{i+1,j-1}$ $-b_{i,j+1}+b_{i,j-1})$	$\frac{1}{h_x^4}(a_{i+1,j})$ $-\frac{1}{16h_x^2h_y^2}(c_{i+1,j+1}+c_{i+1,j-1}$ $+a_{i+1,j+1}+a_{i+1,j-1})$
$j+1$	$-\frac{1}{4h_x^3h_y}$ $(b_{i-1,j}+b_{i-1,j+1})$	$\frac{1}{2h_x^3h_y}(b_{i,j}+b_{i-1,j+1})$ $+\frac{1}{2h_xh_y^3}(b_{i-1,j+1}+b_{i,j})$	$\frac{1}{4h_x^3h_y}(b_{i-1,j}-b_{i+1,j}$ $+b_{i+1,j+1}-b_{i-1,j+1})$ $-\frac{2}{h_y^4}(c_{i,j+1}+c_{i,j})$	$-\frac{1}{2h_x^3h_y}(b_{i,j}+b_{i+1,j+1})$ $-\frac{1}{2h_xh_y^3}(b_{i+1,j+1}+b_{i,j})$	$\frac{1}{4h_x^3h_y}$ $(b_{i+1,j}+b_{i+1,j+1})$
$j+2$	$\frac{1}{16h_x^2h_y^2}$ $(c_{i-1,j+1}+a_{i-1,j+1})$	$-\frac{1}{4h_xh_y^3}$ $(b_{i-1,j+1}+b_{i,j+1})$	$-\frac{1}{16h_x^2h_y^2}(c_{i+1,j+1}+c_{i-1,j+1}$ $+a_{i+1,j+1}+a_{i-1,j+1})$ $+\frac{1}{h_y^4}(c_{i,j+1})$	$\frac{1}{4h_xh_y^3}$ $(b_{i+1,j+1}+b_{i,j+1})$	$\frac{1}{16h_x^2h_y^2}$ $(c_{i+1,j+1}+a_{i+1,j+1})$

Table A.4: Standard stencil for a second order anisotropic regularizer.

A.5 Coupled Regularization Isotropic

	$i-1$	i	$i+1$
$j-1$		$\frac{1}{2h_y^2}(\Psi'_{i,j} + \Psi'_{i,j-1})$	
j	$\frac{1}{2h_x^2}(\Psi'_{i,j} + \Psi'_{i-1,j})$	$-\frac{1}{2h_x^2}(\Psi'_{i+1,j} + 2\Psi'_{i,j} + \Psi'_{i-1,j})$ $-\frac{1}{2h_y^2}(\Psi'_{i,j+1} + 2\Psi'_{i,j} + \Psi'_{i,j-1})$	$\frac{1}{2h_x^2}(\Psi'_{i,j} + \Psi'_{i+1,j})$
$j+1$		$\frac{1}{2h_y^2}(\Psi'_{i,j} + \Psi'_{i,j+1})$	

$j-1$			
j	$\frac{1}{4h_x}(\Psi'_{i,j} + \Psi'_{i-1,j})$		$-\frac{1}{4h_x}(\Psi'_{i+1,j} + \Psi'_{i,j})$
$j+1$			

$j-1$		$\frac{1}{4h_y}(\Psi'_{i,j} + \Psi'_{i,j-1})$	
j			
$j+1$		$-\frac{1}{4h_y}(\Psi'_{i,j+1} + \Psi'_{i,j})$	

Table A.5: Stencils for a coupled isotropic regularizer. **Top:** Stencil used with u or v. **Center:** Stencil used with p_1 or q_1. **Bottom:** Stencil used with p_2 or q_2.

A.6 Coupled Regularization Anisotropic

	$i-1$	i	$i+1$
$j-1$	$\frac{1}{4h_xh_y}(b_{i-1,j}+b_{i,j-1})$	$\frac{1}{2h_y^2}(c_{i,j}+c_{i,j-1})$	$-\frac{1}{4h_xh_y}(b_{i+1,j}+b_{i,j-1})$
j	$\frac{1}{2h_x^2}(a_{i,j}+a_{i-1,j})$	$-\frac{1}{2h_x^2}(a_{i+1,j}+2a_{i,j}+a_{i-1,j})$ $-\frac{1}{2h_y^2}(c_{i,j+1}+2c_{i,j}+c_{i,j-1})$	$\frac{1}{2h_x^2}(a_{i,j}+a_{i+1,j})$
$j+1$	$-\frac{1}{4h_xh_y}(b_{i-1,j}+b_{i,j+1})$	$\frac{1}{2h_y^2}(c_{i,j}+c_{i,j+1})$	$\frac{1}{4h_xh_y}(b_{i+1,j}+b_{i,j+1})$

$j-1$		$\frac{1}{2h_y}(b_{i,j-1})$	
j	$\frac{1}{4h_x}(a_{i,j}+a_{i-1,j})$		$-\frac{1}{4h_x}(a_{i+1,j}+a_{i,j})$
$j+1$		$-\frac{1}{2h_y}(b_{i,j+1})$	

$j-1$		$\frac{1}{4h_y}(c_{i,j}+c_{i,j-1})$	
j	$\frac{1}{2h_x}(b_{i-1,j})$		$-\frac{1}{2h_x}(b_{i+1,j})$
$j+1$		$-\frac{1}{4h_y}(c_{i,j+1}+c_{i,j})$	

Table A.6: Standard stencils for a coupled anisotropic regularizer. **Top:** Stencil used with u or v. **Center:** Stencil used with p_1 or q_1. **Bottom:** Stencil used with p_2 or q_2.

B Derivations

B.1 Second Order Isotropic Discretization

We employ central differences as described in Section 2.3 to approximate the derivatives of second order isotropic regularizers. For a clear derivation we set $\Psi' := \Psi'(\|\mathcal{H}_u\|_{\mathrm{F}}^2 + \|\mathcal{H}_v\|_{\mathrm{F}}^2)$.

$$\operatorname{div}_2\left(D_{2,\mathrm{I}} \cdot \begin{pmatrix} \nabla \mu_x \\ \nabla \mu_y \end{pmatrix}\right) = \partial_{xx}\left(\Psi' \cdot \mu_{xx}\right) + 2 \cdot \partial_{xy}\left(\Psi' \cdot \mu_{xy}\right) + \partial_{yy}\left(\Psi' \cdot \mu_{yy}\right)$$

$$\begin{aligned}
\partial_{xx}\left(\Psi' \cdot \mu_{xx}\right) &\approx \frac{1}{h_x^2} \cdot \Big(\left(\Psi' \cdot \mu_{xx}\right)_{i+1,j} - 2\left(\Psi' \cdot \mu_{xx}\right)_{i,j} + \left(\Psi' \cdot \mu_{xx}\right)_{i-1,j}\Big) \\
&\approx \frac{1}{h_x^2} \cdot \frac{1}{h_x^2} \cdot \Big(\quad \Psi'_{i+1,j}\left(\mu_{i+2,j} - 2\mu_{i+1,j} + \mu_{i,j}\right) \\
&\qquad - 2\Psi'_{i,j} \quad \left(\mu_{i+1,j} - 2\mu_{i,j} + \mu_{i-1,j}\right) \\
&\qquad + \quad \Psi'_{i-1,j}\left(\mu_{i,j} - 2\mu_{i-1,j} + \mu_{i-2,j}\right)\Big) \\
2 \cdot \partial_{xy}\left(\Psi' \cdot \mu_{xy}\right) &\approx \frac{2}{4h_xh_y} \cdot \Big(\quad \left(\Psi' \cdot \mu_{xy}\right)_{i+1,j+1} - \left(\Psi' \cdot \mu_{xy}\right)_{i+1,j-1} \\
&\qquad - \left(\Psi' \cdot \mu_{xy}\right)_{i-1,j+1} + \left(\Psi' \cdot \mu_{xy}\right)_{i-1,j-1}\Big) \\
&\approx \frac{2}{4h_xh_y} \cdot \frac{1}{4h_xh_y} \cdot \Big(\quad \Psi'_{i+1,j+1}\left(\mu_{i+2,j+2} - \mu_{i+2,j} - \mu_{i,j+2} + \mu_{i,j}\right) \\
&\qquad - \Psi'_{i+1,j-1}\left(\mu_{i+2,j} - \mu_{i+2,j-2} - \mu_{i,j} + \mu_{i,j-2}\right) \\
&\qquad - \Psi'_{i-1,j+1}\left(\mu_{i,j+2} - \mu_{i,j} - \mu_{i-2,j+2} + \mu_{i-2,j}\right) \\
&\qquad + \Psi'_{i-1,j-1}\left(\mu_{i,j} - \mu_{i,j-2} - \mu_{i-2,j} + \mu_{i-2,j-2}\right)\Big) \\
\partial_{yy}\left(\Psi' \cdot \mu_{yy}\right) &\approx \frac{1}{h_y^2}\Big(\left(\Psi' \cdot \mu_{yy}\right)_{i,j+1} - 2\left(\Psi' \cdot \mu_{yy}\right)_{i,j} + \left(\Psi' \cdot \mu_{yy}\right)_{i,j-1}\Big) \\
&\approx \frac{1}{h_y^2} \cdot \frac{1}{h_y^2} \cdot \Big(\quad \Psi'_{i,j+1}\left(\mu_{i,j+2} - 2\mu_{i,j+1} + \mu_{i,j}\right) \\
&\qquad - 2\Psi'_{i,j} \quad \left(\mu_{i,j+1} - 2\mu_{i,j} + \mu_{i,j-1}\right) \\
&\qquad + \quad \Psi'_{i,j-1}\left(\mu_{i,j} - 2\mu_{i,j-1} + \mu_{i,j-2}\right)\Big)
\end{aligned}$$

B.2 Second Order Anisotropic Discretization

The same central differences scheme and notation are used as before. In addition, we assign

$$\begin{aligned} a &:= r_{11}^2\Psi_1' + r_{21}^2\Psi_2', \\ b &:= r_{11}r_{12}\Psi_1' + r_{21}r_{22}\Psi_2', \\ c &:= r_{12}^2\Psi_1' + r_{22}^2\Psi_2', \end{aligned} \qquad D_{2,\mathrm{A}} := \begin{pmatrix} a & b & 0 & 0 \\ b & c & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & b & c \end{pmatrix}.$$

Then, the discretization for the second order anisotropic regularizer looks as follows:

$$\begin{aligned} \operatorname{div}_2\left(D_{2,\mathrm{A}} \cdot \begin{pmatrix} \nabla\mu_x \\ \nabla\mu_y \end{pmatrix}\right) = \quad & \partial_{xx}\left(a\cdot\mu_{xx} + b\cdot\mu_{xy}\right) + \partial_{xy}\left(b\cdot\mu_{xx} + c\cdot\mu_{xy}\right) \\ & + \partial_{xy}\left(a\cdot\mu_{xy} + b\cdot\mu_{yy}\right) + \partial_{yy}\left(b\cdot\mu_{xy} + c\cdot\mu_{yy}\right) \end{aligned}$$

$$\begin{aligned} \partial_{xx}\left(a\cdot\mu_{xx}\right) &\approx \frac{1}{h_x^2}\cdot\Big(\left(a\cdot\mu_{xx}\right)_{i+1,j} - 2\left(a\cdot\mu_{xx}\right)_{i,j} + \left(a\cdot\mu_{xx}\right)_{i-1,j}\Big) \\ &\approx \frac{1}{h_x^2}\cdot\frac{1}{h_x^2}\cdot\big(\quad a_{i+1,j}\left(\mu_{i+2,j} - 2\mu_{i+1,j} + \mu_{i,j}\right) \\ &\qquad\qquad - 2a_{i,j}\quad\left(\mu_{i+1,j} - 2\mu_{i,j} + \mu_{i-1,j}\right) \\ &\qquad\qquad + \ a_{i-1,j}\left(\mu_{i,j} - 2\mu_{i-1,j} + \mu_{i-2,j}\right)\Big) \\ \partial_{xx}\left(b\cdot\mu_{xy}\right) &\approx \frac{1}{h_x^2}\cdot\Big(\left(b\cdot\mu_{xy}\right)_{i+1,j} - 2\left(b\cdot\mu_{xy}\right)_{i,j} + \left(b\cdot\mu_{xy}\right)_{i-1,j}\Big) \\ &\approx \frac{1}{h_x^2}\cdot\frac{1}{4h_xh_y}\cdot\big(\quad b_{i+1,j}\left(\mu_{i+2,j+1} - \mu_{i+2,j-1} - \mu_{i,j+1} + \mu_{i,j-1}\right) \\ &\qquad\qquad - 2b_{i,j}\quad\left(\mu_{i+1,j+1} - \mu_{i+1,j-1} - \mu_{i-1,j+1} + \mu_{i-1,j-1}\right) \\ &\qquad\qquad + \ b_{i-1,j}\left(\mu_{i,j+1} - \mu_{i,j-1} - \mu_{i-2,j+1} + \mu_{i-2,j-1}\right)\Big) \\ \partial_{xy}\left(b\cdot\mu_{xx}\right) &\approx \frac{1}{4h_xh_y}\cdot\Big(\quad \left(b\cdot\mu_{xx}\right)_{i+1,j+1} - \left(b\cdot\mu_{xx}\right)_{i+1,j-1} \\ &\qquad\qquad - \left(b\cdot\mu_{xx}\right)_{i-1,j+1} + \left(b\cdot\mu_{xx}\right)_{i-1,j-1}\Big) \\ &\approx \frac{1}{4h_xh_y}\cdot\frac{1}{h_x^2}\cdot\Big(\quad b_{i+1,j+1}\left(\mu_{i+2,j+1} - 2\mu_{i+1,j+1} + \mu_{i,j+1}\right) \\ &\qquad\qquad - b_{i+1,j-1}\left(\mu_{i+2,j-1} - 2\mu_{i+1,j-1} + \mu_{i,j-1}\right) \\ &\qquad\qquad - b_{i-1,j+1}\left(\mu_{i,j+1} - 2\mu_{i-1,j+1} + \mu_{i-2,j+1}\right) \\ &\qquad\qquad + b_{i-1,j-1}\left(\mu_{i,j-1} - 2\mu_{i-1,j-1} + \mu_{i-2,j-1}\right)\Big) \end{aligned}$$

$$
\begin{aligned}
\partial_{xy}\left(c\cdot\mu_{xy}\right) \approx \frac{1}{4h_xh_y}\cdot\Big(\quad & \left(c\cdot\mu_{xy}\right)_{i+1,j+1} - \left(c\cdot\mu_{xy}\right)_{i+1,j-1} \\
& - \left(c\cdot\mu_{xy}\right)_{i-1,j+1} + \left(c\cdot\mu_{xy}\right)_{i-1,j-1}\Big) \\
\approx \frac{1}{4h_xh_y}\cdot\frac{1}{4h_xh_y}\cdot\Big(\quad & c_{i+1,j+1}\left(\mu_{i+2,j+2}-\mu_{i+2,j}-\mu_{i,j+2}+\mu_{i,j}\right) \\
& - c_{i+1,j-1}\left(\mu_{i+2,j}-\mu_{i+2,j-2}-\mu_{i,j}+\mu_{i,j-2}\right) \\
& - c_{i-1,j+1}\left(\mu_{i,j+2}-\mu_{i,j}-\mu_{i-2,j+2}+\mu_{i-2,j}\right) \\
& + c_{i-1,j-1}\left(\mu_{i,j}-\mu_{i,j-2}-\mu_{i-2,j}+\mu_{i-2,j-2}\right)\Big)
\end{aligned}
$$

$$
\begin{aligned}
\partial_{xy}\left(a\cdot\mu_{xy}\right) \approx \frac{1}{4h_xh_y}\cdot\Big(\quad & \left(a\cdot\mu_{xy}\right)_{i+1,j+1} - \left(a\cdot\mu_{xy}\right)_{i+1,j-1} \\
& - \left(a\cdot\mu_{xy}\right)_{i-1,j+1} + \left(a\cdot\mu_{xy}\right)_{i-1,j-1}\Big) \\
\approx \frac{1}{4h_xh_y}\cdot\frac{1}{4h_xh_y}\cdot\Big(\quad & a_{i+1,j+1}\left(\mu_{i+2,j+2}-\mu_{i+2,j}-\mu_{i,j+2}+\mu_{i,j}\right) \\
& - a_{i+1,j-1}\left(\mu_{i+2,j}-\mu_{i+2,j-2}-\mu_{i,j}+\mu_{i,j-2}\right) \\
& - a_{i-1,j+1}\left(\mu_{i,j+2}-\mu_{i,j}-\mu_{i-2,j+2}+\mu_{i-2,j}\right) \\
& + a_{i-1,j-1}\left(\mu_{i,j}-\mu_{i,j-2}-\mu_{i-2,j}+\mu_{i-2,j-2}\right)\Big)
\end{aligned}
$$

$$
\begin{aligned}
\partial_{xy}\left(b\cdot\mu_{yy}\right) \approx \frac{1}{4h_xh_y}\cdot\Big(\quad & \left(b\cdot\mu_{yy}\right)_{i+1,j+1} - \left(b\cdot\mu_{yy}\right)_{i+1,j-1} \\
& - \left(b\cdot\mu_{yy}\right)_{i-1,j+1} + \left(b\cdot\mu_{yy}\right)_{i-1,j-1}\Big) \\
\approx \frac{1}{4h_xh_y}\cdot\frac{1}{h_y^2}\cdot\Big(\quad & b_{i+1,j+1}\left(\mu_{i+1,j+2}-2\mu_{i+1,j+1}+\mu_{i+1,j}\right) \\
& - b_{i+1,j-1}\left(\mu_{i+1,j}-2\mu_{i+1,j-1}+\mu_{i+1,j-2}\right) \\
& - b_{i-1,j+1}\left(\mu_{i-1,j+2}-2\mu_{i-1,j+1}+\mu_{i-1,j}\right) \\
& + b_{i-1,j-1}\left(\mu_{i-1,j}-2\mu_{i-1,j-1}+\mu_{i-1,j-2}\right)\Big)
\end{aligned}
$$

$$
\begin{aligned}
\partial_{yy}\left(b\cdot\mu_{xy}\right) &\approx \frac{1}{h_y^2}\cdot\Big(\left(b\cdot\mu_{xy}\right)_{i,j+1}-2\left(b\cdot\mu_{xy}\right)_{i,j}+\left(b\cdot\mu_{xy}\right)_{i,j-1}\Big)\\
&\approx \frac{1}{h_y^2}\cdot\frac{1}{4h_xh_y}\cdot\big(\quad b_{i,j+1}\left(\mu_{i+1,j+2}-\mu_{i+1,j}-\mu_{i-1,j+2}+\mu_{i-1,j}\right)\\
&\qquad -2b_{i,j}\quad\left(\mu_{i+1,j+1}-\mu_{i+1,j-1}-\mu_{i-1,j+1}+\mu_{i-1,j-1}\right)\\
&\qquad +\ b_{i,j-1}\left(\mu_{i+1,j}-\mu_{i+1,j-2}-\mu_{i-1,j}+\mu_{i-1,j-2}\right)\Big)\\
\partial_{yy}\left(c\cdot\mu_{yy}\right) &\approx \frac{1}{h_y^2}\cdot\Big(\left(c\cdot\mu_{yy}\right)_{i,j+1}-2\left(c\cdot\mu_{yy}\right)_{i,j}+\left(c\cdot\mu_{yy}\right)_{i,j-1}\Big)\\
&\approx \frac{1}{h_y^2}\cdot\frac{1}{h_y^2}\cdot\big(\quad c_{i,j+1}\left(\mu_{i,j+2}-2\mu_{i,j+1}+\mu_{i,j}\right)\\
&\qquad -2c_{i,j}\quad\left(\mu_{i,j+1}-2\mu_{i,j}+\mu_{i,j-1}\right)\\
&\qquad +\ c_{i,j-1}\left(\mu_{i,j}-2\mu_{i,j-1}+\mu_{i,j-2}\right)\Big)
\end{aligned}
$$

B.3 Coupled Isotropic Discretization

A nested central difference scheme with half the step sizes is used in the following to discretize the divergence expression with $\Psi' := \Psi'(|\nabla u - \mathbf{p}|^2 + |\nabla v - \mathbf{q}|^2)$.

$$\operatorname{div}\left(D_{\mathrm{C,I}} \cdot (\nabla u - \mathbf{p})\right) = \partial_x \left(\Psi' \cdot (u_x - p_1)\right) + \partial_y \left(\Psi' \cdot (u_y - p_2)\right)$$

$$\begin{aligned}
\partial_x \left(\Psi' \cdot (u_x - p_1)\right) &\approx 2 \cdot \frac{1}{2h_x} \cdot \left(\left(\Psi' \cdot (u_x - p_1)\right)_{i+\frac{1}{2},j} - \left(\Psi' \cdot (u_x - p_1)\right)_{i-\frac{1}{2},j}\right) \\
&\approx \quad 2 \cdot \frac{1}{2h_x} \cdot \left(\frac{\Psi'_{i+1,j} + \Psi'_{i,j}}{2} \cdot \left(\frac{u_{i+1,j} - u_{i,j}}{2\frac{1}{2}h_x} - \frac{(p_1)_{i+1,j} + (p_1)_{i,j}}{2}\right)\right) \\
&\quad - 2 \cdot \frac{1}{2h_x} \cdot \left(\frac{\Psi'_{i,j} + \Psi'_{i-1,j}}{2} \cdot \left(\frac{u_{i,j} - u_{i-1,j}}{2\frac{1}{2}h_x} - \frac{(p_1)_{i,j} + (p_1)_{i-1,j}}{2}\right)\right) \\
&= \quad \frac{\Psi'_{i+1,j} + \Psi'_{i,j}}{2h_x^2} \cdot \left(u_{i+1,j} - u_{i,j}\right) - \frac{\Psi'_{i+1,j} + \Psi'_{i,j}}{4h_x} \cdot \left((p_1)_{i+1,j} + (p_1)_{i,j}\right) \\
&\quad - \frac{\Psi'_{i,j} + \Psi'_{i-1,j}}{2h_x^2} \cdot \left(u_{i,j} - u_{i-1,j}\right) + \frac{\Psi'_{i,j} + \Psi'_{i-1,j}}{4h_x} \cdot \left((p_1)_{i,j} + (p_1)_{i-1,j}\right) \\
\partial_y \left(\Psi' \cdot (u_y - p_2)\right) &\approx 2 \cdot \frac{1}{2h_y} \cdot \left(\left(\Psi' \cdot (u_y - p_2)\right)_{i,j+\frac{1}{2}} - \left(\Psi' \cdot (u_y - p_2)\right)_{i,j-\frac{1}{2}}\right) \\
&\approx \quad 2 \cdot \frac{1}{2h_y} \cdot \left(\frac{\Psi'_{i,j+1} + \Psi'_{i,j}}{2} \cdot \left(\frac{u_{i,j+1} - u_{i,j}}{2\frac{1}{2}h_y} - \frac{(p_2)_{i,j+1} + (p_2)_{i,j}}{2}\right)\right) \\
&\quad - 2 \cdot \frac{1}{2h_y} \cdot \left(\frac{\Psi'_{i,j} + \Psi'_{i,j-1}}{2} \cdot \left(\frac{u_{i,j} - u_{i,j-1}}{2\frac{1}{2}h_y} - \frac{(p_2)_{i,j} + (p_2)_{i,j-1}}{2}\right)\right) \\
&= \quad \frac{\Psi'_{i,j+1} + \Psi'_{i,j}}{2h_y^2} \cdot \left(u_{i,j+1} - u_{i,j}\right) - \frac{\Psi'_{i,j+1} + \Psi'_{i,j}}{4h_y} \cdot \left((p_2)_{i,j+1} + (p_2)_{i,j}\right) \\
&\quad - \frac{\Psi'_{i,j} + \Psi'_{i,j-1}}{2h_y^2} \cdot \left(u_{i,j} - u_{i,j-1}\right) + \frac{\Psi'_{i,j} + \Psi'_{i,j-1}}{4h_y} \cdot \left((p_2)_{i,j} + (p_2)_{i,j-1}\right)
\end{aligned}$$

The same discretization scheme can be applied to $\operatorname{div}\left(D_{\mathrm{C,I}} \cdot (\nabla v - \mathbf{q})\right)$.

B.4 Coupled Anisotropic Discretization

Again nested central differences are used to approximate the derivatives of the divergence expression using the assignment

$$\begin{aligned} a &:= r_{11}^2\Psi_1' + r_{21}^2\Psi_2', \\ b &:= r_{11}r_{12}\Psi_1' + r_{21}r_{22}\Psi_2', \\ c &:= r_{12}^2\Psi_1' + r_{22}^2\Psi_2', \end{aligned} \qquad D_{\mathrm{C,A}} := \begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

where $\Psi_\star' := \Psi_\star'((\mathbf{r}_\star^\top(\nabla u - \mathbf{p}))^2 + (\mathbf{r}_\star^\top(\nabla v - \mathbf{q}))^2)$ and $\star \in \{1, 2\}$.

$$\operatorname{div}(D_{\mathrm{C,A}} \cdot (\nabla u - \mathbf{p})) = \partial_x \left(a \cdot (u_x - p_1) + b \cdot \left(u_y - p_2\right)\right) + \partial_y \left(b \cdot (u_x - p_1) + c \cdot \left(u_y - p_2\right)\right)$$

The derivatives $\partial_x(a \cdot (u_x - p_1))$ and $\partial_y(c \cdot (u_y - p_2))$ can be determined as in Section B.3.

$$\begin{aligned} \partial_x \left(b \cdot \left(u_y - p_2\right)\right) &\approx \frac{1}{2h_x} \cdot \left(\left(b \cdot \left(u_y - p_2\right)\right)_{i+1,j} - \left(b \cdot \left(u_y - p_2\right)\right)_{i-1,j}\right) \\ &\approx \quad \frac{1}{2h_x} \cdot \left(b_{i+1,j} \cdot \left(\frac{u_{i+1,j+1} - u_{i+1,j-1}}{2h_y} - (p_2)_{i+1,j}\right)\right) \\ &\quad - \frac{1}{2h_x} \cdot \left(b_{i-1,j} \cdot \left(\frac{u_{i-1,j+1} - u_{i-1,j-1}}{2h_y} - (p_2)_{i-1,j}\right)\right) \\ &= \quad \frac{b_{i+1,j}}{4h_x h_y} \cdot \left(u_{i+1,j+1} - u_{i+1,j-1}\right) - \frac{b_{i+1,j}}{2h_x} \cdot (p_2)_{i+1,j} \\ &\quad - \frac{b_{i-1,j}}{4h_x h_y} \cdot \left(u_{i-1,j+1} - u_{i-1,j-1}\right) + \frac{b_{i-1,j}}{2h_x} \cdot (p_2)_{i-1,j} \\ \partial_y \left(b \cdot (u_x - p_1)\right) &\approx \frac{1}{2h_y} \cdot \left(\left(b \cdot (u_x - p_1)\right)_{i,j+1} - \left(b \cdot (u_x - p_1)\right)_{i,j-1}\right) \\ &\approx \quad \frac{1}{2h_y} \cdot \left(b_{i,j+1} \cdot \left(\frac{u_{i+1,j+1} - u_{i-1,j+1}}{2h_x} - (p_1)_{i,j+1}\right)\right) \\ &\quad - \frac{1}{2h_y} \cdot \left(b_{i,j-1} \cdot \left(\frac{u_{i+1,j-1} - u_{i-1,j-1}}{2h_x} - (p_1)_{i,j-1}\right)\right) \\ &= \quad \frac{b_{i,j+1}}{4h_x h_y} \cdot \left(u_{i+1,j+1} - u_{i-1,j+1}\right) - \frac{b_{i,j+1}}{2h_y} \cdot (p_1)_{i,j+1} \\ &\quad - \frac{b_{i,j-1}}{4h_x h_y} \cdot \left(u_{i+1,j-1} - u_{i-1,j-1}\right) + \frac{b_{i,j-1}}{2h_y} \cdot (p_1)_{i,j-1} \end{aligned}$$

The same discretization scheme can be applied to $\operatorname{div}(D_{\mathrm{C,A}} \cdot (\nabla v - \mathbf{q}))$.

Bibliography

[1] P. Moulin, R. Krishnamurthy, and J. W. Woods. “Multiscale modeling and estimation of motion fields for video coding”. In: *IEEE Transactions on Image Processing* 6.12 (1997), pp. 1606–1620.

[2] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. “High accuracy optical flow estimation based on a theory for warping”. In: *Proc. European Conference on Computer Vision (ECCV)*. Vol. 3024. Springer, 2004, pp. 25–36.

[3] J. Xiong, H. Li, Q. Wu, and F. Meng. “A fast HEVC inter CU selection method based on pyramid motion divergence”. In: *IEEE Transactions on Multimedia* 16.2 (2014), pp. 559–564.

[4] K. Souhila and A. Karim. “Optical flow based robot obstacle avoidance”. In: *International Journal of Advanced Robotic Systems* 4.1 (2007), pp. 13–16.

[5] A. Fernández-Caballero, J. C. Castillo, J. Martínez-Cantos, and R. Martínez-Tomás. “Optical flow or image subtraction in human detection from infrared camera on mobile robot”. In: *Robotics and Autonomous Systems* 58.12 (2010), pp. 1273–1281.

[6] D. Pfeiffer and U. Franke. “Efficient representation of traffic scenes by means of dynamic stixels”. In: *Proc. IEEE Intelligent Vehicles Symposium (IV)*. 2010, pp. 217–224.

[7] A. Geiger, P. Lenz, and R. Urtasun. “Are we ready for autonomous driving? The KITTI vision benchmark suite”. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. `http://www.cvlibs.net/datasets/kitti/`. 2012, pp. 3354–3361.

[8] A. Bruhn. “Variational Optic Flow Computation: Accurate Modelling and Efficient Numerics”. Dissertation. Saarbrücken: Department of Mathematics and Computer Science, Saarland University, 2006.

[9] A. Singh. “An estimation-theoretic framework for image-flow computation”. In: *Proc. IEEE International Conference on Computer Vision (ICCV)*. 1990, pp. 168–177.

[10] H. G. Musmann, P. Pirsch, and H.-J. Grallert. “Advances in picture coding”. In: *Proceedings of the IEEE* 73.4 (1985), pp. 523–548.

[11] P. H. S. Torr and A. Zisserman. “Feature based methods for structure and motion estimation”. In: *Proc. International Workshop on Vision Algorithms*. Springer. 1999, pp. 278–294.

[12] J. Wills, S. Agarwal, and S. Belongie. “A feature-based approach for dense segmentation and estimation of large disparity motion”. In: *International Journal of Computer Vision* 68.2 (2006), pp. 125–143.

[13] B. Lucas and T. Kanade. "An iterative image registration technique with an application to stereo vision." In: *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*. Vol. 81. 1. 1981, pp. 674–679.

[14] B. Horn and B. Schunck. "Determining optical flow". In: *Artificial Intelligence* 17.1-3 (1981), pp. 185–203.

[15] H. Zimmer, A. Bruhn, J. Weickert, L. Valgaerts, A. Salgado, B. Rosenhahn, and H.-P. Seidel. "Complementary optic flow". In: *Proc. International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*. Springer. 2009, pp. 207–220.

[16] S. Uras, F. Girosi, A. Verri, and V. Torre. "A computational approach to motion perception". In: *Biological Cybernetics* 60.2 (1988), pp. 79–87.

[17] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. "Highly accurate optic flow computation with theoretically justified warping". In: *International Journal of Computer Vision* 67.2 (2006), pp. 141–158.

[18] O. Demetz, D. Hafner, and J. Weickert. "The Complete Rank Transform: A tool for accurate and morphologically invariant matching of structures". In: *British Machine Vision Conference (BMVC)*. 2013.

[19] M. J. Black and P. Anandan. "Robust dynamic motion estimation over time". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1991, pp. 296–302.

[20] A. Bruhn and J. Weickert. "Towards ultimate motion estimation: combining highest accuracy with real-time performance". In: *Proc. IEEE International Conference on Computer Vision (ICCV)*. Vol. 1. 2005, pp. 749–755.

[21] L. I. Rudin, S. Osher, and E. Fatemi. "Nonlinear total variation based noise removal algorithms". In: *Physica D: Nonlinear Phenomena* 60.1 (1992), pp. 259–268.

[22] A. Chambolle and P.-L. Lions. "Image recovery via total variation minimization and related problems". In: *Numerische Mathematik* 76.2 (1997), pp. 167–188.

[23] O. Demetz, M. Stoll, S. Volz, J. Weickert, and A. Bruhn. "Learning brightness transfer functions for the joint recovery of illumination changes and optical flow". In: *Proc. European Conference on Computer Vision (ECCV)*. Springer. 2014, pp. 455–471.

[24] W. Trobin, T. Pock, D. Cremers, and H. Bischof. "An unbiased second-order prior for high-accuracy motion estimation". In: *Proc. German Conference on Pattern Recognition*. Springer. 2008, pp. 396–405.

[25] K. Bredies, K. Kunisch, and T. Pock. "Total Generalized Variation". In: *SIAM Journal on Imaging Sciences* 3.3 (2010), pp. 492–526.

[26] D. Hafner, C. Schroers, and J. Weickert. "Introducing maximal anisotropy into second order coupling models". In: *Proc. German Conference on Pattern Recognition*. Springer. 2015, pp. 79–90.

[27] F. Lenzen, F. Becker, and 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*. Springer. 2013, pp. 61–73.

[28] H. Zimmer, A. Bruhn, and J. Weickert. “Optic flow in harmony”. In: *International Journal of Computer Vision* 93.3 (2011), pp. 368–388.

[29] W. Förstner and E. Gülch. “A fast operator for detection and precise location of distinct points, corners and centres of circular features”. In: *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*. 1987, pp. 281–305.

[30] D. Maurer. “Depth-Driven Variational Methods for Stereo Reconstruction”. Master Thesis. Stuttgart: Institute of Visualization and Interactive Systems, University of Stuttgart, 2014.

[31] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. “Performance of optical flow techniques”. In: *International Journal of Computer Vision* 12.1 (1994), pp. 43–77.

[32] A. Bruhn, J. Weickert, and Ch. Schnörr. “Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods”. In: *International Journal of Computer Vision* 61.3 (2005), pp. 211–231.

[33] S. Baker, D. Scharstein, J.P. Lewis, S. Roth, M.J. Black, and R. Szeliski. “A database and evaluation methodology for optical flow”. In: *International Journal of Computer Vision* 92.1 (2011). http://vision.middlebury.edu/flow/data/, pp. 1–31.

[34] A. N. Tikhonov. “Solution of incorrectly formulated problems and the regularization method”. In: *Soviet Mathematics Doklady*. Vol. 5. 1963, pp. 1035–1038.

[35] S. Didas, J. Weickert, and B. Burgeth. “Properties of higher order nonlinear diffusion filtering”. In: *Journal of Mathematical Imaging and Vision* 35.3 (2009), pp. 208–226.

[36] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. “Two deterministic half-quadratic regularization algorithms for computed imaging”. In: *Proc. IEEE International Conference on Image Processing (ICIP)*. Vol. 2. 1994, pp. 168–172.

[37] O. Scherzer and J. Weickert. “Relations between regularization and diffusion filtering”. In: *Journal of Mathematical Imaging and Vision* 12.1 (2000), pp. 43–63.

[38] P. Perona and 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.

[39] A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr. “A multigrid platform for real-time motion computation with discontinuity-preserving variational methods”. In: *International Journal of Computer Vision* 70.3 (2006), pp. 257–277.

[40] D. Young. “Iterative methods for solving partial difference equations of elliptic type”. In: *Transactions of the American Mathematical Society* 76.1 (1954), pp. 92–111.

[41] D. Sun, S. Roth, J.P. Lewis, and M.J. Black. “Learning optical flow”. In: *Proc. European Conference on Computer Vision (ECCV)*. Springer, 2008, pp. 83–97.

[42] J. Weickert. *Anisotropic Diffusion in Image Processing*. Vol. 1. Teubner Stuttgart, 1998.

[43] J. A. Nelder and R. Mead. “A simplex method for function minimization”. In: *The Computer Journal* 7.4 (1965), pp. 308–313.

[44] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. “A naturalistic open source movie for optical flow evaluation”. In: *Proc. European Conference on Computer Vision (ECCV)*. `http://sintel.is.tue.mpg.de/`. Springer, Oct. 2012, pp. 611–625.

[45] M. Menze and A. Geiger. “Object scene flow for autonomous vehicles”. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. `http://www.cvlibs.net/datasets/kitti/`. 2015, pp. 3061–3070.

[46] S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer. “Modeling temporal coherence for optical flow”. In: *Proc. IEEE International Conference on Computer Vision (ICCV)*. 2011, pp. 1116–1123.

All links were last visited and found working on April 7, 2017.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature