Adaptive Algorithms for 3D Reconstruction and Motion Estimation

Von der Fakultät für Informatik, Elektrotechnik und Informationstechnik der Universität Stuttgart zur Erlangung der Würde eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

Daniel Rudolf Maurer

aus Rochester Hills, Michigan,
Vereinigte Staaten von Amerika

Hauptberichter:	Prof. Dr. Andrés Bruhn
Mitberichter:	Prof. Dr. Thomas Brox
Tag der mündlichen Prüfung:	24.10.2019

Institut für Visualisierung und Interaktive Systeme (VIS)
der Universität Stuttgart

2019

Prüfungsausschuss

Vorsitzender:	Prof. Dr. sc. ETH Andreas Bulling, Universität Stuttgart
Hauptberichter:	Prof. Dr.-Ing. Andrés Bruhn, Universität Stuttgart
Mitberichter:	Prof. Dr.-Ing. Thomas Brox, Albert-Ludwigs-Universität Freiburg
Mitprüfer:	Prof. Dr. rer. nat. Marc Toussaint, Universität Stuttgart

Abstract

The number of applications influenced by computer vision has increased rapidly in the last few years. Innovative technologies such as autonomous robotic vacuum cleaners, smart video doorbells, and augmented reality devices use image data to sense the surrounding environment. On a basic level, sensing the environment can mean to recover the 3D geometry of the real world and the objects contained therein as well as to capture the motion field between consecutive image frames of a video sequence. In this thesis, we deal with these two challenges that constitute two fundamental problems in computer vision: *3D reconstruction* and *motion estimation*. In particular we aim at improving upon the accuracy of current methods by utilizing adaptive approaches as well as by considering multiple sources of information.

In the first part of the thesis, we deal with the topic of *3D reconstruction*. We develop and investigate an approach that leverages multiple depth cues simultaneously. In particular, we combine two cues that complement each other: the parallax cue and the shading cue. While the parallax cue (stereo) allows to obtain accurate estimates in highly textured regions, the shading cue (shape from shading) allows to improve the reconstruction in homogeneous areas. Furthermore, we formulate the model in such a way that it not only estimates the shape but simultaneously computes the albedo and the illumination. This formulation renders the method especially adaptive regarding the adaptation to different scenes, objects, and illumination conditions. To complete the new model we employ special anisotropic smoothness terms. This in turn enables a detail-preserving regularization. Regarding the optimization we propose a novel hyperbolic warping scheme based on an upwind approximation. This new scheme nicely blends in with the commonly used geometric warping and hence allows for a convenient optimization of the proposed model. As a result, we obtain an adaptive method that enables the estimation of high-quality depth maps of Lambertian scenes with varying albedo under unknown illumination. The results clearly demonstrate the advantages over single cue based approaches as well as the capability to recover fine surface details.

In the second part of the thesis, we turn to the topic of *motion estimation*. In this context, our contributions concern order-adaptive regularization strategies, advanced refinement models for pipeline approaches and multi-frame strategies for pipeline approaches. First, we analyze and compare different isotropic and anisotropic second-order regularization strategies for variational motion estimation that allow the computation of accurate affine motion fields. In this context, we propose a new order-adaptive approach that brings together first and second-order regularization within a single model that combines the benefits of both techniques. Second, we design a new model for variational refinement that tackles the shortcomings of current refinement models. In particular, it combines an illumination-aware data term that offers robustness under varying illumination with our novel order-adaptive regularization strategy that is capable to estimate accurate affine flow fields. Third, we propose two different techniques to leverage additional input frames within the motion estimation: a strategy based on an ego-motion model and a strategy based on a learned motion model. Both techniques enable us to significantly improve the estimation results, especially, in case of out of frame motion and in case of occluded areas. Overall, starting from a variational approach with fixed-order regularization we succeed to steadily improve the results, finally obtaining state-of-the-art quality on the most popular benchmark data sets.

Zusammenfassung

Die Zahl der Anwendungen, die von Computer Vision beeinflusst sind, ist in den letzten Jahren rasant gestiegen. Innovative Technologien wie autonome Saugroboter, smarte Video-Türklingeln und Augmented-Reality-Geräte greifen vermehrt auf Bilddaten zurück, um ihre Umgebung zu erfassen. Der erste Schritt dafür, kann daraus bestehen die 3D-Geometrie der realen Welt und die darin enthaltenen Objekte zu rekonstruieren oder die Bewegung in Bildabfolgen zu messen. In dieser Arbeit beschäftigen wir uns mit genau diesen beiden fundamentalen Problemstellungen des Themengebiets: *3D Rekonstruktion* und *Bewegungsbestimmung*. Insbesondere zielen wir darauf ab, die Genauigkeit von derzeitigen Verfahren zu verbessern, indem wir adaptive Algorithmen entwickeln, die es ermöglichen mehrere Informationsquellen simultan auszunutzen.

Im ersten Teil der Arbeit beschäftigen wir uns mit 3D Rekonstruktion. Wir entwickeln und untersuchen ein Verfahren, welches mehrere Prinzipien zur Tiefenwahrnehmung kombiniert. Dabei greifen wir auf zwei sich ergänzende Prinzipien zurück: Parallaxe und Schattierung. Während die Parallaxe genaue Messungen in stark texturierten Bereichen ermöglicht, erlaubt die Schattierung präsize Messungen in homogenen Bereichen. Darüber hinaus formulieren wir das Modell, dass es nicht nur die Form berechnet, sondern auch die Albedo und die Beleuchtung. Diese Formulierung macht die Methode besonders adaptiv hinsichtlich der Anpassung an verschiedenste Szenen, Objekte und Lichtverhältnisse. Um das neue Modell abzurunden, verwenden wir spezielle anisotrope Glattheitsterme. Dies wiederum ermöglicht eine detail-erhaltende Regularisierung. Im Hinblick auf die Optimierung stellen wir ein neues hyperbolisches Warping-Schema vor, das auf einer Upwind-Approximation basiert. Dieses neue Schema fügt sich harmonisch in das verbreitete geometrische Warping-Schema ein und ermöglicht eine geeignete Optimierung des Modells. Als Ergebnis erhalten wir eine adaptive Methode, die die Berechnung von präzisen Tiefenkarten von Lambertschen Szenen mit variierenden Albedo unter unbekannter Beleuchtung ermöglicht. Die Ergebnisse zeigen deutlich die Vorteile gegenüber Verfahren, die nur einzelne Prinzipien zur Tiefenwahrnehmung verwenden, sowie die Fähigkeit, feine Oberflächendetails zu erfassen.

Im zweiten Teil der Arbeit widmen wir uns der Bewegungsbestimmung. In diesem Kontext umfassen unsere Beiträge ordnungsadaptive Regularisierungsstrategien, Refinement-Modelle für Pipeline-Methoden, sowie Mehrbildstrategien für Pipeline-Methoden. Zuerst analysieren und vergleichen wir isotrope und anisotrope Regularisiererstrategien zweiter Ordnung für variationelle Bewegungsbestimmung, die die genaue Berechnung von affinen Bewegungsfeldern ermöglichen. In diesem Zusammenhang entwerfen wir einen neuen ordnungsadaptiven Ansatz, der die Regularisierung erster und zweiter Ordnung in einem einzigen Modell vereint und die Vorteile beider Techniken kombiniert. Zweitens stellen wir ein neues Refinement-Modell vor, das die Schwächen von aktuellen Modellen behebt. Insbesondere kombiniert es einen speziellen Datenterm, der Robustheit bei Beleuchtungsänderungen bietet, mit unserem neuartigen ordnungsadaptiven Regularisier, der in der Lage ist, genaue affine Bewegungsfelder zu berechnen. Drittens stellen wir zwei Techniken vor, die Informationen aus mehr als zwei Bildern gewinnen: basierend auf einem Eigen-Bewegungsmodell sowie einem gelernten Bewegungsmodell. Beide Strategien ermöglichen signifikante Verbesserungen, insbesondere im Fall von Verdeckungen. Letzendlich gelingt es uns, ausgehend von einem Variationsansatz mit festgelegter Regularisierungsordnung, die Verfahren stetig zu verbessern und schlussendlich Ergebnisse zu produzieren die dem aktuellen Stand der Technik auf den gängigsten Benchmark-Datensätzen darstellen.

Acknowledgements

At this point, I would like to thank all the people that supported me in doing this work. Without them, the completion of my dissertation would not have been possible. First of all, I would like to express my deepest appreciation to Andrés Bruhn for giving me the opportunity to be a part of his group and for supervising me throughout my dissertation. It was a great pleasure working together and I really appreciate all the valuable advice and support I received. Second, I am also grateful to Thomas Brox for being the second reviewer of this thesis.

Further I would like to thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for support within Project B04 "Adaptive Algorithms for Motion Estimation" of SFB/Transregio 161 "Quantitative Methods for Visual Computing" (Project Number 251654672) and within the joint Project BR 2245/3-1 and BR 4372/1-1 "Variationsmethoden zur Fusion von Shape from Shading und Stereo" (Project Number 214118604).

I also want to thank all my colleagues from the Institute for Visualization and Interactive Systems as well as from the SFB/Transregio 161 for providing a pleasant working atmosphere. In particular many thanks to all the former and current members of our group: Sebastian Volz, Yong Chul Ju, Michael Stoll, Simon Rühle, and Azin Jahedi. I really enjoyed being part of this group. In this context, special thanks to Yong Chul Ju for our fruitful cooperations on shape from shading and to Michael Stoll for our great collaborations on optical flow. Moreover, I would like to acknowledge the assistance of the system administrators, Anton Malina and Martin Schmid, for taking care of computer related challenges and the secretaries, Margot Roubicek and Christine Schütz, for their uncomplicated and friendly help with organizational aspects. I would also like to extend my sincere thanks to Kai Mindermann for accompanying me throughout my bachelor, master, and doctoral studies and the great fun we had during this time.

Finally, I would also like to extend my deepest gratitude to my entire family for their love as well as their unconditional and endless support. I truly feel blessed to have my parents Heinrich and Marion, my brother David, and my grandparents Rudolf and Heidemarie. Most importantly, I am extremely grateful to my wife Damaris for her warmth and love, and my son Ian for bringing endless time of joy in my life. I am incredibly fortunate to have you in my life.

“<qXY4XB b w eqw4e sxdyx3yyyyyyyyyy<xĝv312ˆ
4CR65Z<Q RG 5″

— Ian Daniel Maurer, 2019

Contents

1 Introduction

Nowadays, the impact of computer vision on our day-to-day lives steadily increases. Meanwhile, computer vision based technology can be found in areas such as retail, automotive, healthcare, agriculture, banking, and many more. For many people, the automatic extraction, analysis, and understanding of captured images and video sequences has already become a matter of course. However, while visually sensing the world might be an easy task for a human being, it is a highly non-trivial task to transfer this capability to a machine. Consequently, decades of research have been necessary to reach the current state-of-the-art.

As in all scientific fields, one can break down the overarching field in smaller subdomains and identify certain key problems. Two such fundamental problems of computer vision are *3D reconstruction* and *motion estimation*. While the goal of 3D reconstruction is to capture/recover the shape of real objects that are lost during the acquisition process, the goal of motion estimation is to determine the displacement field between frame pairs in an image sequence. In this thesis, we aim at advancing the state-of-the-art in both subdomains. To this end, we not only develop essential concepts that allow extracting multiple sources of information simultaneously but also propose mechanisms that allow adapting to the underlying data. Furthermore, we realize implementations of these concepts and mechanisms either in terms of individual steps of larger estimation pipelines or in terms of standalone approaches.

1.1 Scope and Contributions

Next, we give a more detailed overview of the problems we consider in the thesis. In particular, we touch on the topics of the different chapters and highlight our contributions.

1.1.1 3D Reconstruction

The first problem we approach is 3D reconstruction. In particular, we focus on passive image-based reconstruction techniques. In contrast to active approaches, such methods do not directly interfere with the scene, e.g., in terms of varying the illumination or using active sensor technology such as time-of-flight cameras, and instead they only operate on standard images. Depending on the underlying strategy these techniques require either a single image or multiple images captured from different viewpoints to perform the reconstruction. Hence, the overall reconstruction process for such approaches typically involves a camera calibration step as well as possible post-processing steps, e.g., point-cloud fusion or mesh generation. However, in this thesis, we concentrate on the reconstruction process and therefore assume the camera setup to be calibrated.

Variational 3D Reconstruction Variational methods represent a very prominent class of techniques in computer vision. Such methods minimize a so-called cost or energy func-

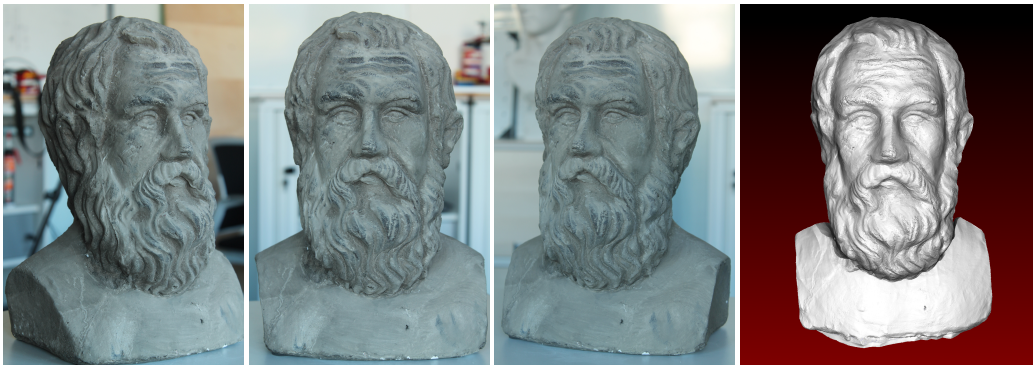


Figure 1.1: Example 3D Reconstruction. *From left to right*: (a-c) Input images [218]. (d) Shaded reconstruction result of our method.

tional, which constitutes a measure of correctness concerning certain assumptions, to solve a specific task, e.g., 3D reconstruction. Typically the formulation of such an energy functional comprises two types of components: data terms and regularization terms. While data terms express certain constraints that characterize the unknowns and relate them to the given data, regularization terms impose some kind of spatial regularity, i.e., smoothness, on the unknowns. In this thesis we advance the field by proposing such a variational approach that simultaneously exploits two fundamentally different depth cues, i.e., the shading cue and the parallax cue. By combining both depth cues, we are able to obtain a robust estimation of the overall object surface due to the parallax cue, while also being able to recover fine surface details due to the shading cue; cf. Figure 1.1. Furthermore, to maximize the applicability of our method, we estimate not only the depth but also the color and reflectance properties (albedo) as well as the present illumination. In this context, the careful selection of the regularization plays an important role. While we use specially tailored anisotropic (directional-dependent) first-order smoothness terms that provide sharp illumination and albedo maps, we employ an anisotropic second-order smoothness term for the depth that allows the reconstruction of slanted surfaces. As a result, we obtain a method that enables the estimation of high-quality depth maps of Lambertian scenes with varying albedo under unknown illumination. Moreover, we not only propose a new variational model but also provide novel ideas for the numerical minimization. In particular, we propose a coarse-to-fine minimization scheme based on a linearization of all data terms and an upwind scheme approximation of the shape from shading data term that allows us to embed the entire optimization into a hierarchical incremental fixed point strategy. Finally, we evaluate our method on commonly used data sets to demonstrate the excellent performance. Furthermore, we compare the results of our approach to the results of approaches that only rely on the parallax cue to show the usefulness as well as the importance of the shading cue to recover fine surface details. Main parts of this work are published in [3, 4, 5].

1.1.2 Motion Estimation

The second problem we tackle is motion estimation. While the term motion estimation is quite generic, we refer to it as the 2D motion field between frame pairs in an image sequence. Further,

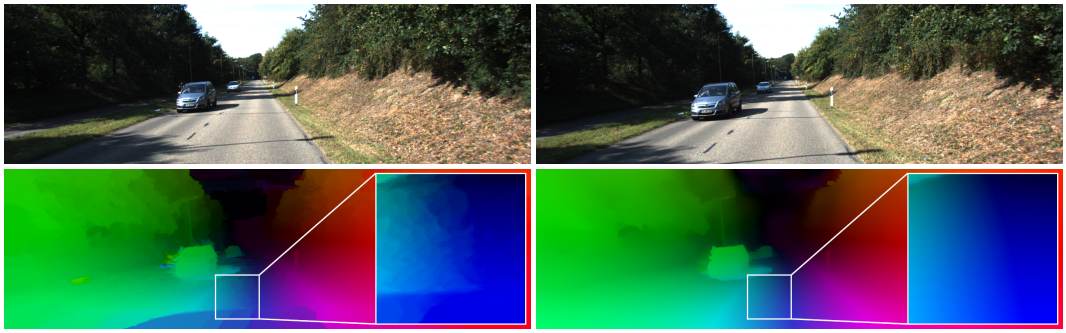


Figure 1.2: Different regularization order for motion estimation. *Top to bottom, from left to right*: (a-b) Input images [119]. (c) Motion field computed using a first-order regularizer. (d) Motion field computed using a second-order regularizer.

we assume that this 2D motion field is the result of the projection of a 3D motion onto the image plane. In the literature, this scenario is also one possible interpretation of the optical flow problem, which is often used interchangeably [174].

Variational Motion Estimation Variational methods also have a long and successful history in the context of motion estimation. In the original formulation of Horn and Schunck [81], the underlying energy functional is composed of two terms: a data term that imposes temporal constancy constraints on image features and a regularization term that enforces spatial regularity on the solution. While the data term enables to trace corresponding points in subsequent frames, the regularization term allows coping with ill-posed situations and to, therefore, obtain a plausible per pixel solution. So far, many variational optical flow methods rely on first-order regularization strategies [36, 124, 162, 193, 216]. Recently, however, approaches based on second-order regularization have gained more and more attention [33, 54, 80, 137, 167]. In particular in scenes with a vast amount of ego-motion, such second-order regularizers allow to estimate the resulting piecewise affine flow fields which cannot be captured adequately by first-order regularizers; cf. Figure 1.2. In this context, we first compare different techniques to model such second-order regularization strategies and demonstrate how to incorporate directional information to steer the underlying smoothing behavior. Second, we propose a new order-adaptive regularization strategy that automatically adapts the regularization order to match the underlying data. This new order-adaptive regularizer enables us to combine the advantages of both first and second-order regularization strategies, i.e., the robustness regarding small fluctuations of first-order regularization and the capability to handle affine motion patterns of second-order regularization. Finally, our evaluation shows that the proposed strategy facilitates generalization across different data sets without the need to manually adjust the underlying model. Main parts of this work are published in [9, 10].

Beyond Variational Motion Estimation Entirely variational methods for motion estimation, as mentioned before, have a well-known weakness – the estimation of large displacements of small objects. To deal with this problem, researchers proposed different strategies [37, 143, 199]. Of these strategies, primarily, the idea to replace the coarse-to-fine minimization scheme by a proper initialization, obtained via a sparse-to-dense interpolation of point correspondences

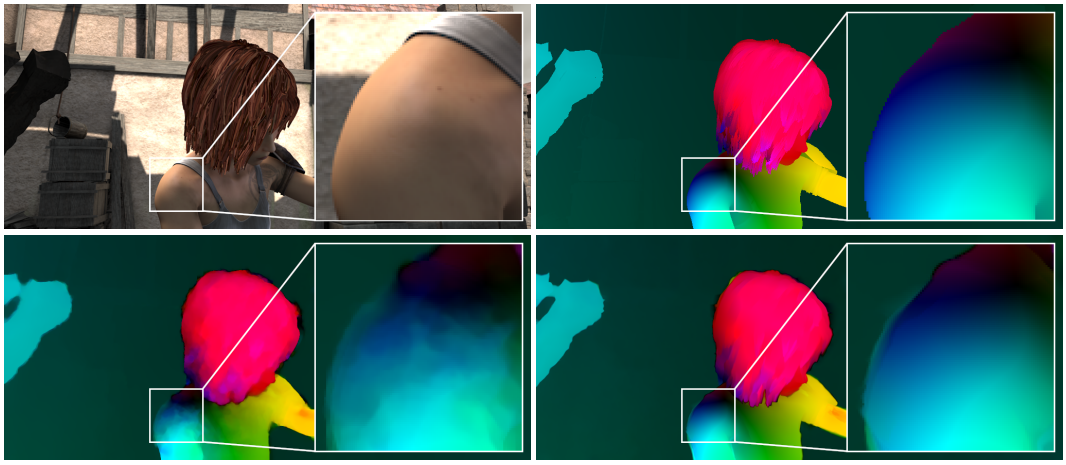


Figure 1.3: Different refinement methods. *Top to bottom, from left to right*: (a-b) Reference input image [44] and ground truth motion field. (c) Motion field computed using a commonly used refinement method [143]. (d) Motion field computed using our refinement method [8].

[143], has prevailed. In fact, most state-of-the-art large displacement optical flow pipelines use it and refer to the variational component as variational refinement [20, 50, 61, 85, 120].

Even though the variational refinement plays an essential role in many recent approaches, most of these new pipeline based methods rely on rather simple models for the refinement. Thus the refinement typically cannot keep up with the adaptivity and robustness of the preceding pipeline steps, which may lead to imprecise motion fields; cf. Figure 1.3. To tackle this shortcoming, we propose a new model for variational refinement that combines robustness under varying illumination with the adaptive estimation of higher-order motion fields. Moreover, we suggest a reduced coarse-to-fine scheme: a hierarchical minimization approach that can benefit from a proper initialization within the pipeline approach while still being able to correct errors in the intermediate results. Finally, the conducted evaluation makes the benefits of our advanced model explicit. In particular, our new refinement scheme consistently improves the results across all major motion estimation benchmarks. Main parts of this work are published in [8].

Multi-Frame Motion Estimation So far, we focused on two-frame methods, i.e., methods that only use two frames of the image sequence. This choice, however, prevents us from exploiting a possibly valuable source of information – information on temporal coherence. Therefore, we propose and investigate strategies that enable us to exploit information from additional input frames. To realize these strategies we extend pipeline based approaches to integrate the following steps: compute the motion w.r.t. preceding frames of the image sequence, relate this additional motion information to the current motion field in terms of predictions, and include these predictions in the estimation process to improve the results; cf. Figure 1.4. To this end, we employ two different motion models that allow us to relate motion fields via predictions: an ego-motion model and a learned motion model. Main parts of this work are published in [2, 6].

In the case of the ego-motion model based approach, we first design a coarse-to-fine multi-frame PatchMatch approach for estimating structure matches (structure from motion) that combines a

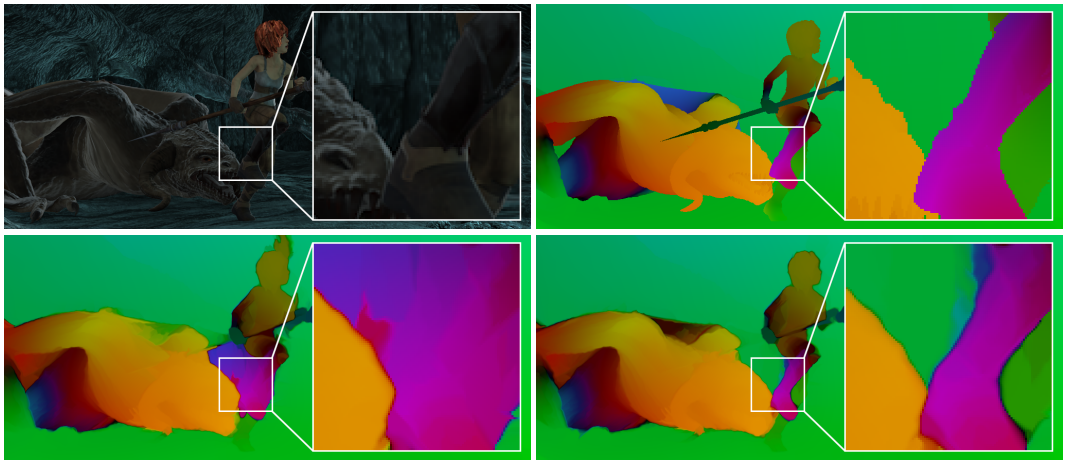


Figure 1.4: Two-Frame and Multi-Frame results. *Top to bottom, from left to right*: (a-b) Reference input image [44] and ground truth motion field. (c) Motion field computed using a two-frame method. (d) Motion field computed using a multi-frame method [2].

depth based parametrization with different temporal selection strategies. While the parametrization models the estimation more robust by reducing the search space, the hierarchical optimization and the temporal selection improve the accuracy. Second, we propose a consistency-based selection scheme for combining predictions from this structure-based PatchMatch approach with matches of an unconstrained PatchMatch approach. Thereby, the backward flow allows us to identify reliable structure matches, while a robust voting scheme decides on the remaining cases. Third, we embed the resulting matches into the optical flow pipeline. By employing recent approaches for interpolation and refinement, our method provides dense results with sub-pixel accuracy. Finally, experiments on all major benchmarks demonstrate the benefits of our novel approach. In particular, the greatest benefits are achieved in the case of occlusions and out-of-frame motion.

Probably the main drawbacks of the ego-motion model are that benefits are limited to rigid parts of the sequence and that sufficient ego-motion is required to work well. To tackle these shortcomings, we are the first to propose a method that relies on a learned motion model that is capable to overcome these limitations. This learned motion model is implemented via a convolutional neural network. In contrast to other approaches that train networks before the estimation, our approach learns the models online, i.e., during the estimation. Moreover, instead of relying on potentially unsuitable data sets with ground truth, our models are trained using initial flow estimates of the actual sequence. Such an unsupervised/self-supervised training offers the advantage that we can learn appropriate models for each sequence. In addition, our approach not only learns one model per sequence but one model for each frame of every sequence. This per-frame learning results in a high degree of adaptability when it comes to a change of the scene content. Finally, the learned models are spatially variant, i.e., location dependent. This ability, in turn, addresses the problem of independently moving objects. At last, we demonstrate within our evaluation that this novel strategy is capable to overcome the shortcomings of the ego-motion model and achieve improvements in the context of independently moving objects, non-ego motion scenes and non-rigid-motion

scenarios. By overcoming these shortcomings it is able to achieve state-of-the-art results on major optical flow benchmarks and is currently among the most accurate methods for motion estimation.

1.2 Outline

First, we cover essential foundations concerning the image formation process as well as the design and optimization of variational models in Chapter 2. Then we tackle the problem of variational 3D reconstruction in Chapter 3 by combining parallax and shading cues within a joint variational approach. Subsequently, we turn towards the topic of motion estimation. Starting with entirely variational methods in Chapter 4, we first look at higher order regularization strategies and propose a new order-adaptive regularization strategy. To overcome certain limitations of entirely variational methods we propose a new refinement model for pipeline based motion estimation methods in Chapter 5. Finally, we propose two different strategies that allow to exploit information from more than two input frames in Chapter 6 and conclude in Chapter 7.

2 Foundations

In this chapter, we introduce basic concepts that are important throughout the entire thesis. It includes ideas related to the image formation process, such as camera geometry and radiometric models, as well as concepts regarding optimization techniques.

2.1 Images

Digital images constitute the source of information for all approaches presented in this thesis. In the case of grayscale images, storing is realized via a two-dimensional array. Within this array, each array element represents a single picture element (pixel) that encodes an intensity value. Further, we interpret every pixel as a discrete sample point of a continuous function $I : \Omega \to \mathbb{R}$, where $\Omega \subset \mathbb{R}^2$ denotes the rectangular image domain. In particular, the sample points are arranged on a regular grid as shown in Figure 2.1, where h_x and h_y denote the horizontal and vertical grid spacing and n_x, n_y denote the number sample points in both directions, respectively. Therefore, the value of a pixel at the array element (i, j) corresponds to the sampling point at

$$\mathbf{x} = \left((i - \tfrac{1}{2}) \cdot h_x, (j - \tfrac{1}{2}) \cdot h_y\right)^\top . \tag{2.1}$$

In the case of RGB color images, a third dimension is added to store the intensities of the red, green and blue color channel. Consequently, the associated function $\mathbf{I} : \Omega \to \mathbb{R}^3$ is vector-valued.

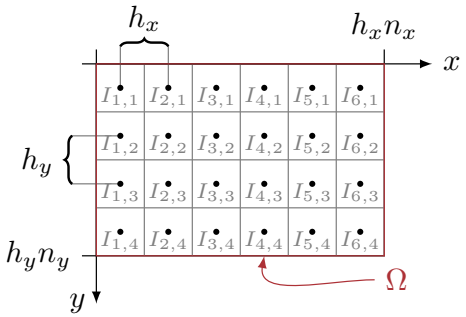


Figure 2.1: Sample points on a regular grid within the rectangular image domain.

2.2 Camera Geometry

Next, we will look into the geometric-related concepts of the image formation process. While the explanations of the introduced concepts are kept rather short and straightforward, Hartley and Zisserman give a more extensive introduction with detailed descriptions in their book [79].

2.2.1 Homogeneous Coordinates

Dealing with the concept of projection, homogeneous coordinates (also termed projective coordinates) turn out to be a useful tool. In particular, they allow to express affine transformations and protective transformations as a single matrix multiplication. To go from Euclidean space $\mathbb{R}^n$ to projective space $\mathbb{P}^n$ an additional dimension is introduced. Therefore, the forward transformation of a point $\mathbf{x} \in \mathbb{R}^n$ to its homogeneous counterpart $\tilde{\mathbf{x}} \in \mathbb{P}^n$ is given by

$$\begin{aligned} \Pi : \mathbb{R}^n &\to \mathbb{P}^n \\ \mathbf{x} = (x_1, \dots, x_n)^\top &\mapsto \tilde{\mathbf{x}} = (x_1, \dots, x_n, 1)^\top , \end{aligned} \tag{2.2}$$

which simply appends a one. The backward transformations reads

$$\begin{aligned} \pi : \mathbb{P}^n &\to \mathbb{R}^n \\ \tilde{\mathbf{x}} = (\tilde{x}_1, \dots, \tilde{x}_n, \tilde{x}_{n+1})^\top &\mapsto \mathbf{x} = \left(\frac{\tilde{x}_1}{\tilde{x}_{n+1}}, \dots, \frac{\tilde{x}_n}{\tilde{x}_{n+1}} \right)^\top , \end{aligned} \tag{2.3}$$

where the first n entries are divided by the last entry $\tilde{x}_{n+1}$ and the additionally introduced dimension is removed. In case of the two-dimensional space $\mathbb{R}^2$ this can be interpreted as expressing 2D points via lines in a 3D space, where all points on the line $\lambda\tilde{\mathbf{x}} = (\lambda x_1, \lambda x_2, \lambda)^\top$ represent the same point $\mathbf{x} = (x_1, x_2)^\top$. Furthermore, all these parallel lines intersect in points at infinity, which have a zero entry in the additional dimension. These points at infinity do not have an Euclidean counterpart and consequently the back transformation π is not defined for $\lambda = 0$.

2.2.2 Pinhole Camera Model

To describe the mapping of a 3D point onto a 2D point on the image plane, a relationship between both points must be defined. The pinhole camera model represents such a relationship, i.e., a perfect perspective projection. Assuming that the camera coordinate system is aligned with the world coordinate system, i.e., the camera center $\mathbf{C}$ coincides with the origin of the world coordinate system, the image plane $\Omega \subset \mathbb{R}^2$ is defined to lie parallel to the X-Y-plane at distance of the focal length f. Figure 2.2 depicts this setup. Here, the Z-axis coincides with the principal axis, which is perpendicular to the image plane and passes through the camera center $\mathbf{C}$. Furthermore, the intersection point $\mathbf{p}$ of the principal axis and the image plane is called the principal point.

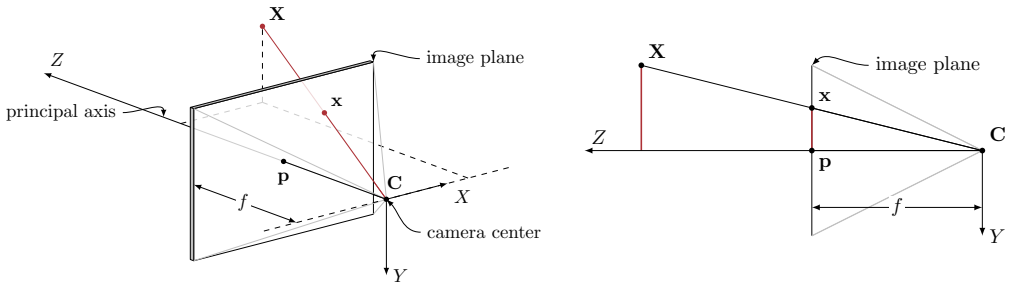


Figure 2.2: Geometry of the pinhole camera model.

The pinhole camera model maps a 3D point $\mathbf{X} \in \mathbb{R}^3$ to the 2D location $\mathbf{x} \in \Omega$ on the image plane, where the line joining the point $\mathbf{X}$ and the camera center $\mathbf{C}$, i.e., the optical ray, intersects the image plane. Figure 2.2 shows this mapping. According to the intercept theorem

$$\frac{x}{f} = \frac{X}{Z} \qquad \text{and} \qquad \frac{y}{f} = \frac{Y}{Z} \tag{2.4}$$

holds such that the projection is given by

$$\mathbf{X} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \cdot \frac{X}{Z} \\ f \cdot \frac{Y}{Z} \end{pmatrix}. \tag{2.5}$$

By employing the previously introduced homogeneous coordinates, the projection can be written in a compact form by using a single matrix multiplication

$$\mathbf{x} = \pi(\tilde{\mathbf{x}}) = \pi\big(P\tilde{\mathbf{X}}\big) \qquad \text{with} \qquad P = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{2.6}$$

where π is the backward transformation of the homogeneous coordinates, see Equation 2.3, and the matrix P is the so-called camera projection matrix.

Intrinsic Camera Parameters The model is generalized to include a principal point offset as well as an individual scaling in both axial directions, to copy the internal characteristics of actual cameras. While adding a principal point offset allows to describe a mapping w.r.t. an image coordinate system that is not aligned with the camera coordinate system, see Figure 2.3, an individual scaling in both axial directions allows to define the mapping w.r.t. a pixel coordinate system. Both generalizations are included via the camera calibration matrix, given by

$$K = \begin{pmatrix} s_x & 0 & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{2.7}$$

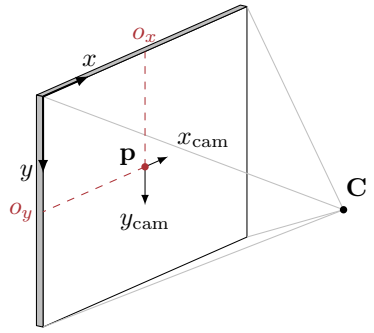


Figure 2.3: Principal point offset $\mathbf{o}$.

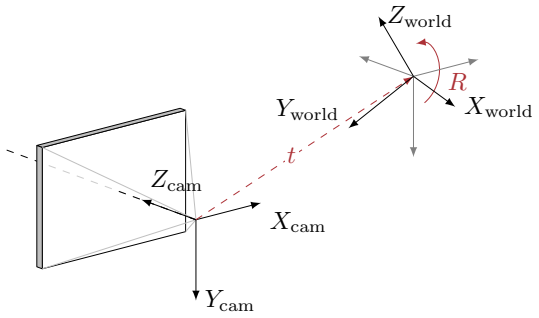


Figure 2.4: Static transformation $T_{\text{cam}\leftarrow\text{world}}$.

where $\mathbf{o} = (o_x, o_y)^\top$ is the principal point offset, and $s_x = f \cdot m_x$ and $s_y = f \cdot m_y$ are the scaled focal length in terms of pixel dimensions in both axial directions, respectively. Here $\mathbf{m} = (m_x, m_y)^\top$ denotes the individual scaling factor that defines the number of pixels per unit length in the image coordinate system and, as before, f is the focal length.

The camera calibration matrix K is an upper triangular matrix with the determinant $\det(K) = s_x \cdot s_y \cdot 1 \neq 0$ and hence it is invertible. The inverse of K reads

$$K^{-1} = \begin{pmatrix} \frac{1}{s_x} & 0 & -\frac{o_x}{s_x} \\ 0 & \frac{1}{s_y} & -\frac{o_y}{s_y} \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.8}$$

Extrinsic Camera Parameters Up to now, the camera coordinate system is assumed to be aligned with the world coordinate system. While this simplification does not pose a problem if only a single camera has to be considered, it does not allow to deal with multiple camera scenarios. To overcome this simplification, we introduce a static transformation that maps between the world coordinate frame and the camera coordinate frame. A rotation and a translation define this static transformation. Using a 3×3 rotation matrix R and a translation 3-vector $\mathbf{t}$, see Figure 2.4, it relates a point $\mathbf{X}^{\text{world}}$ defined in the world coordinate frame to the equivalent point $\mathbf{X}^{\text{cam}}$ defined in the camera coordinate frame by

$$\mathbf{X}^{\text{cam}} = R\,\mathbf{X}^{\text{world}} + \mathbf{t}\,. \tag{2.9}$$

This transformation can also be encoded in a 4×4 matrix

$$T_{\text{cam}\leftarrow\text{world}} = \begin{pmatrix} R & \mathbf{t} \\ 0 & 1 \end{pmatrix}, \tag{2.10}$$

which allows applying the static transformation via a single matrix-vector product by using homogeneous coordinates

$$\tilde{\mathbf{X}}^{\text{cam}} = T_{\text{cam}\leftarrow\text{world}}\,\tilde{\mathbf{X}}^{\text{world}}\,. \tag{2.11}$$

The inverse transformation $T^{-1}_{\text{cam}\leftarrow\text{world}} := T_{\text{world}\leftarrow\text{cam}}$ is given by

$$T_{\text{world}\leftarrow\text{cam}} = \begin{pmatrix} R^{-1} & -R^{-1}\mathbf{t} \\ 0 & 1 \end{pmatrix}, \tag{2.12}$$

where $R^{-1} = R^{\top}$, since the rotation matrix R is an orthogonal matrix, and where $\mathbf{C}^{\text{world}} = -R^{-1}\mathbf{t}$ is the camera center in terms of the world coordinate frame.

General Pinhole Camera Model Finally, combining the intrinsic camera parameters and the extrinsic camera parameters allows describing the mapping from a 3D point $\mathbf{X}^{\text{world}}$ onto the corresponding 2D point $\mathbf{x}$ on the image plane

$$\mathbf{x} = \pi(\tilde{\mathbf{x}}) = \pi(K\,(R\,\mathbf{X}^{\text{world}} + \mathbf{t}))\,. \tag{2.13}$$

Making further use of homogeneous coordinates allows describing the projection by the so-called camera projection matrix P, given by

$$P = (K\ 0)\,T_{\text{cam}\leftarrow\text{world}} = K\,(R\ t) \tag{2.14}$$

that maps a homogeneous 3D point defined in the world coordinate frame $\tilde{\mathbf{X}}^{\text{world}}$ to the corresponding 2D pixel location in the image coordinate system via

$$\mathbf{x} = \pi(\tilde{\mathbf{x}}) = \pi(P\,\tilde{\mathbf{X}}^{\text{world}})\,. \tag{2.15}$$

In total, the presented general camera projection matrix offers ten degrees of freedom, of which four arise from the camera calibration matrix K, and the remaining six are due to the rotation and translation encoded in the extrinsic camera parameters.

2.2.3 Back Projection on the Surface

The previous section described the projection of a 3D point onto the image plane. Now the inverse operation shall be discussed, i.e., the back projection. This back projection comes down to a parametrization of the optical ray that passes through the camera center $\mathbf{C} = -R^{-1}\mathbf{t}$ and a point $\mathbf{x}$ on the image plane. By considering the projection described in Equation 2.13, a 3D point $\mathbf{X}$ that lies on the optical ray is given by

$$\begin{aligned} \tilde{\mathbf{x}} &= KR\mathbf{X} + K\mathbf{t} \\ \tilde{\mathbf{x}} - K\mathbf{t} &= KR\mathbf{X} \\ K^{-1}\tilde{\mathbf{x}} - \mathbf{t} &= R\mathbf{X} \\ R^{-1}K^{-1}\tilde{\mathbf{x}} - R^{-1}\mathbf{t} &= \mathbf{X} \\ R^{-1}K^{-1}\tilde{\mathbf{x}} + \mathbf{C} &= \mathbf{X}\,. \end{aligned} \tag{2.16}$$

Therefore, the optical ray can be parametrized as

$$\begin{aligned}\mathbf{s}(\mathbf{x}, z) &= \mathbf{C} + z \cdot (\mathbf{X} - \mathbf{C}) \\ &= \mathbf{C} + z \cdot R^{-1} K^{-1} \tilde{\mathbf{x}}\,, \end{aligned} \tag{2.17}$$

where z is the distance of the resulting point $\mathbf{s}(\mathbf{x}, z)$ to the camera center $\mathbf{C}$. By definition, the distance z is measured along the principal axis. In case the world coordinate frame is aligned with the camera coordinate frame the back projection reduces to

$$\mathbf{s}(\mathbf{x}, z) = z \cdot K^{-1}\tilde{\mathbf{x}} = z \cdot \begin{pmatrix} \frac{1}{s_x} & 0 & -\frac{o_x}{s_x} \\ 0 & \frac{1}{s_y} & -\frac{o_y}{s_y} \\ 0 & 0 & 1 \end{pmatrix} \tilde{\mathbf{x}} = z \cdot \begin{pmatrix} \frac{x - o_x}{s_x} \\ \frac{y - o_y}{s_y} \\ 1 \end{pmatrix}. \tag{2.18}$$

In general 3D points lie on a surface. The corresponding surface normal at a certain point can be computed as the normalized cross-product

$$\mathbf{n}(\mathbf{x}, z) = \frac{\partial_x \mathbf{s} \times \partial_y \mathbf{s}}{|\partial_x \mathbf{s} \times \partial_y \mathbf{s}|}\,, \tag{2.19}$$

of the corresponding tangent vectors, given by the partial derivatives

$$\partial_x \mathbf{s} = R^{-1} K^{-1} (z_x \tilde{\mathbf{x}} + z\,\mathbf{e}_1)\,, \tag{2.20}$$

$$\partial_y \mathbf{s} = R^{-1} K^{-1} (z_y \tilde{\mathbf{x}} + z\,\mathbf{e}_2)\,, \tag{2.21}$$

where $\mathbf{e}_1 = (1, 0, 0)^\top$ and $\mathbf{e}_2 = (0, 1, 0)^\top$. The cross product can be computed as follows

$$\begin{aligned}\bar{\mathbf{n}}(\mathbf{x}, z) &= \partial_x \mathbf{s} \times \partial_y \mathbf{s} \\ &= R^{-1} K^{-1} (z_x \tilde{\mathbf{x}} + z\,\mathbf{e}_1) \times R^{-1} K^{-1} (z_y \tilde{\mathbf{x}} + z\,\mathbf{e}_2)\,. \end{aligned} \tag{2.22}$$

Using the following three algebraic properties

$$(M\mathbf{a}) \times (M\mathbf{b}) = (\det M) M^{-\top} (\mathbf{a} \times \mathbf{b})\,, \tag{2.23}$$

$$\mathbf{a} \times \mathbf{a} = \mathbf{0}\,, \tag{2.24}$$

$$\mathbf{a} \times \mathbf{b} = -(\mathbf{b} \times \mathbf{a})\,, \tag{2.25}$$

where M denotes a 3×3 matrix and $\mathbf{a}$ and $\mathbf{b}$ are two 3-vectors, allows simplifying the expression

$$\begin{aligned}\bar{\mathbf{n}}(\mathbf{x}, z) &= M \left((z_x \tilde{\mathbf{x}} + z\,\mathbf{e}_1) \times (z_y \tilde{\mathbf{x}} + z\,\mathbf{e}_2) \right) \\ &= M \left((z_x \tilde{\mathbf{x}}) \times (z_y \tilde{\mathbf{x}} + z\,\mathbf{e}_2) + (z\,\mathbf{e}_1) \times (z_y \tilde{\mathbf{x}} + z\,\mathbf{e}_2) \right) \\ &= M \left(z_x z_y (\tilde{\mathbf{x}} \times \tilde{\mathbf{x}}) + z_x z (\tilde{\mathbf{x}} \times \mathbf{e}_2) + z_y z (\mathbf{e}_1 \times \tilde{\mathbf{x}}) + z^2 (\mathbf{e}_1 \times \mathbf{e}_2) \right) \\ &= z \cdot M \left(z_x (\tilde{\mathbf{x}} \times \mathbf{e}_2) - z_y (\tilde{\mathbf{x}} \times \mathbf{e}_1) + z \mathbf{e}_3 \right) \end{aligned}$$

$$
\begin{aligned}
&= z \cdot M \left(\begin{pmatrix} -z_x \\ 0 \\ x \end{pmatrix} + \begin{pmatrix} 0 \\ -z_y \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ z \end{pmatrix} \right) \\
&= z \cdot M \begin{pmatrix} -z_x \\ -z_y \\ z_x x + z_y y + z \end{pmatrix} \\
&= z \cdot M \begin{pmatrix} -z_x \\ -z_y \\ \nabla z^\top \mathbf{x} + z \end{pmatrix}
\end{aligned} \tag{2.26}
$$

with $\mathbf{e}_3 = (0, 0, 1)^\top$ and $M = \det(R^{-1}K^{-1})(R^{-1}K^{-1})^{-\top} = \frac{1}{s_x s_y} R^\top K^\top$. Finally, replacing M in the expression leads to

$$
\bar{\mathbf{n}}(\mathbf{x}, z) = \frac{z}{s_x s_y} R^\top K^\top \begin{pmatrix} -z_x \\ -z_y \\ \nabla z^\top \mathbf{x} + z \end{pmatrix} . \tag{2.27}
$$

Again in the special case that the world coordinate frame is aligned with the camera coordinate frame, the cross product reads

$$
\bar{\mathbf{n}}(\mathbf{x}, z) = \frac{z}{s_x s_y} K^\top \begin{pmatrix} -z_x \\ -z_y \\ \nabla z^\top \mathbf{x} + z \end{pmatrix} = \frac{z}{s_x s_y} \begin{pmatrix} -s_x z_x \\ -s_y z_y \\ \nabla z^\top (\mathbf{x} - \mathbf{o}) + z \end{pmatrix} , \tag{2.28}
$$

such that

$$
\mathbf{n}(\mathbf{x}, z) = \frac{\bar{\mathbf{n}}(\mathbf{x}, z)}{|\bar{\mathbf{n}}(\mathbf{x}, z)|} = \frac{\begin{pmatrix} -s_x z_x \\ -s_y z_y \\ \nabla z^\top (\mathbf{x} - \mathbf{o}) + z \end{pmatrix}}{\sqrt{s_x^2 z_x^2 + s_y^2 z_y^2 + (\nabla z^\top (\mathbf{x} - \mathbf{o}) + z)^2}} . \tag{2.29}
$$

Eventually, one should note that the surface normal may either point inwards or outwards the actual object. Therefore, $\mathbf{n}(\mathbf{x}, z)$, as well as $-\mathbf{n}(\mathbf{x}, z)$, impose a valid surface normal. In our case, we have a right-handed coordinate system, such that the surface normal pointing outwards reads

$$
\mathbf{n}(\mathbf{x}, z) = -\frac{\bar{\mathbf{n}}(\mathbf{x}, z)}{|\bar{\mathbf{n}}(\mathbf{x}, z)|} . \tag{2.30}
$$

2.2.4 Epipolar Geometry

The previously described pinhole camera model allows specifying the mapping of 3D points onto a 2D image plane. With two cameras that capture the scene from two distinct locations, the corresponding projections obey certain geometric constraints, known as epipolar constraints. Figure 2.5 depicts such a two-camera setup. The line connecting both camera centers $\mathbf{C}$ and $\mathbf{C}'$ is the baseline. The intersections of the baseline and the image planes define the epipoles $\mathbf{e}$ and $\mathbf{e}'$, respectively.

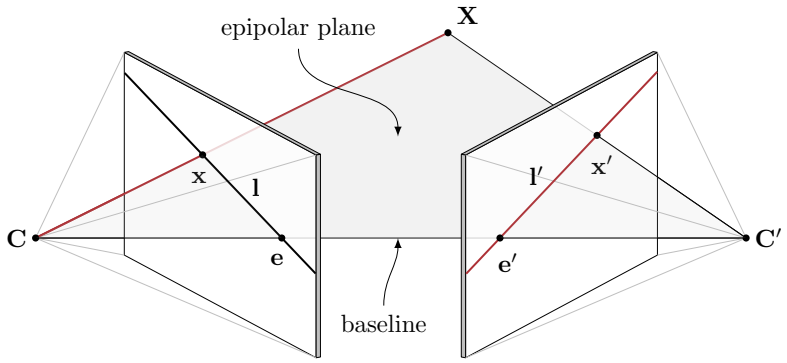


Figure 2.5: Sketch showing the epipolar geometry.

Furthermore, all planes that contain the baseline are epipolar planes, which intersect the image planes in the so-called epipolar lines $\mathbf{l}$ and $\mathbf{l}'$, respectively.

Epipolar Constraint Given a point $\mathbf{x}$ on the image plane, the corresponding scene point $\mathbf{X}$ must lie on the optical ray going through $\mathbf{x}$ and the camera center $\mathbf{C}$. The projection of this optical ray onto the image plane of another camera $\mathbf{C}'$ results in the epipolar line $\mathbf{l}'$. Consequently, the corresponding projection $\mathbf{x}'$ of the scene point $\mathbf{X}$ cannot be arbitrary and must lie on the epipolar line $\mathbf{l}'$. This restriction is known as the epipolar constraint which we can formalize as

$$\tilde{\mathbf{x}}'^{\top}\mathbf{l}' = 0\,. \tag{2.31}$$

In order to compute the epipolar line, one can calculate the cross product between two homogeneous points that lie on it, e.g.,

$$\mathbf{l}' = \mathbf{e}' \times \tilde{\mathbf{x}}'\,. \tag{2.32}$$

2.3 Radiometric Model

After detailing on the geometric part of the image formation process in terms of the pinhole camera model, this section covers the relevant radiometric parts. As before the provided information is kept rather brief and for more details, we refer to the book of Glassner [68].

2.3.1 Basic Radiometric Quantities

In the following, we give a brief overview of relevant fundamental radiometric quantities.

- *Radiant energy*, Q, is the energy traveling in electromagnetic waves.
- *Radiant flux* or *radiant power*, Φ, is the time rate of change of the radiant energy

$$\Phi = \frac{\partial Q}{\partial t}\,, \tag{2.33}$$

  where t denotes the time.

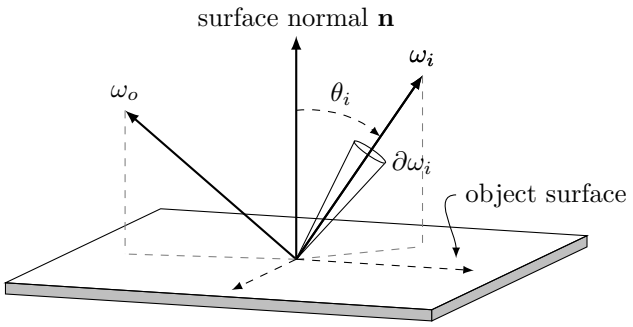


Figure 2.6: Geometry related to the bidirectional reflectance distribution function (BRDF).

- *Radiant flux density*, u, is the quotient of the radiant flux on or emitted by a differential surface element ∂A at a point, divided by the area of the element

$$u = \frac{\partial \Phi}{\partial A} \,. \tag{2.34}$$

  While the radiant flux density incident on a surface is called *irradiance*, E, the radiant flux density emitted by a surface is called *radiant exitance*.

- *Radiance*, L, is the radiant flux per unit projected area perpendicular to the ray per unit solid angle in the direction of the ray

$$L = \frac{\partial^2 \Phi}{\partial A \, \cos\theta \, \partial\omega} \,. \tag{2.35}$$

  It is a convenient and fundamental radiometric quantity associated with a light ray. On the one hand, it remains constant as it propagates along a direction, assuming a vacuum. On the other hand, all other radiometric quantities can be derived from it.

All the previously listed radiometric quantities are functions of wavelength, time, position, direction and polarization. However, by suppressing any dependence on polarization, assuming that the energy of different wavelengths is decoupled and no time-depended behavior is present, i.e., light travels infinitely fast, the terms solely depend on a position $\mathbf{X}$ and a direction ω, e.g., $L(\mathbf{X}, \omega)$.

2.3.2 Bidirectional Reflectance Distribution Function

The reflection of light of a surface is not only proportional to the incoming light but also depends on the surface reflectance properties. To characterize this proportionality the bidirectional reflectance distribution function (BRDF) is used

$$f_r(\mathbf{X}, \omega_i, \omega_o) = \frac{\partial L_r(\mathbf{X}, \omega_o)}{\partial E_i(\mathbf{X}, \omega_i)} = \frac{\partial L_r(\mathbf{X}, \omega_o)}{L_f(\mathbf{X}, \omega_i) \, \cos\theta_i \, \partial\omega_i} \tag{2.36}$$

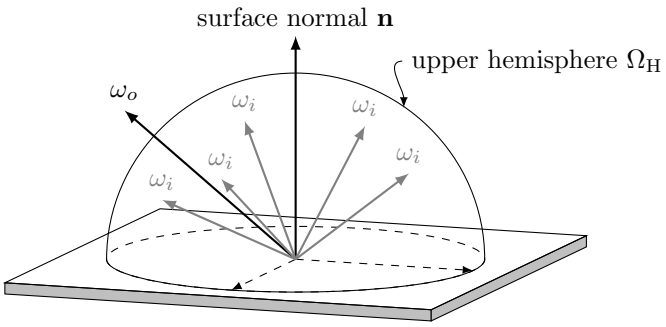


Figure 2.7: Geometry related to the rendering equation.

where E_i is the surface irradiance, L_f is the field radiance and L_r is the reflected radiance. Figure 2.6 shows a sketch of the related geometry. Physically plausible BRDFs must fulfill the Helmholtz reciprocity principle and uphold the law of conservation of energy. While the first requirement just means that the outcome of the BRDF is not affected if the incident and reflected directions are swapped

$$f_r(\mathbf{X}, \omega_i, \omega_o) = f_r(\mathbf{X}, \omega_o, \omega_i)\,, \tag{2.37}$$

the second requirement states that the outgoing radiance must be less or equal to the incoming radiance, such that by integrating over the upper hemisphere Ω_H the following holds

$$\int_{\Omega_\mathrm{H}} f_r(\mathbf{X}, \omega_i, \omega_o)\, \cos\theta_i\, \partial\omega_i \;\leq\; 1\,. \tag{2.38}$$

2.3.3 The Rendering Equation

Rewriting the previously introduced BRDF allows expressing the reflected radiance in terms of the incoming radiance from a single ray and the BRDF associated with the surface point $\mathbf{X}$

$$\partial L_r(\mathbf{X}, \omega_o) = f_r(\mathbf{X}, \omega_i, \omega_o) L_f(\mathbf{X}, \omega_i)\, \cos\theta_i\, \partial\omega_i\,. \tag{2.39}$$

Now, by integrating over the upper hemisphere Ω_H, see Figure 2.7, one obtains the total reflected radiance at the surface point $\mathbf{X}$ in direction ω_o

$$L_o(\mathbf{X}, \omega_o) = \int_{\Omega_\mathrm{H}} f_r(\mathbf{X}, \omega_i, \omega_o) L_i(\mathbf{X}, \omega_i) \cos\theta_i\, d\omega_i\,, \tag{2.40}$$

where $\cos\theta_i = \omega_i \cdot \mathbf{n}$ can be computed as the dot product of the direction ω_i and the surface normal $\mathbf{n}$. Finally, by further considering the emitted radiance $L_e(\mathbf{X}, \omega_o)$ we obtain the so-called rendering equation [90, 97]

$$L_o(\mathbf{X}, \omega_o) = L_e(\mathbf{X}, \omega_o) + \int_{\Omega_\mathrm{H}} f_r(\mathbf{X}, \omega_i, \omega_o) L_i(\mathbf{X}, \omega_i)(\omega_i \cdot \mathbf{n})\, d\omega_i\,. \tag{2.41}$$

2.3.4 Lambertian Reflectance

Finally, let us introduce the reflectance property that we will consider within this thesis – the Lambertian reflectance. It describes an ideal diffuse reflection, i.e., reflected light scatters in all possible directions over the upper hemisphere. Furthermore, it is viewpoint independent, i.e., independent from a viewing direction. Hence, the associated BRDF is constant and reads

$$f_{\text{Lambertian}} = \frac{\rho}{\pi}\,. \tag{2.42}$$

2.4 Variational Modeling

So far, we introduced basic geometric and radiometric concepts that describe the overall image formation process. Next, we turn to the topic of variational modeling, where we explain how we can formalize the considered computer vision problems in such a way, that we can solve it on a machine. This formalization includes the development of a measure of goodness of the alternatives, typically described by a so-called objective or cost function. By minimizing or maximizing this objective function, one obtains one of the best solutions from all feasible solutions.

Throughout this thesis, *functionals* will constitute different objective functions. Therefore, the following section starts with a brief introduction to the calculus of variations, which is concerned with the extrema of functionals. Gelfand and Fomin give a more in-depth treatment of the topic in their book [66].

2.4.1 Calculus of Variations

The calculus of variations is concerned with the extrema of *functionals*. A functional can be regarded as a function of functions since it assigns a scalar to each function belonging to a particular class. The following general form can express most of the functionals considered in this thesis

$$E(\mathbf{u}) = \int_{\Omega} F(\mathbf{x}, u_1, \dots, u_n, \nabla u_1, \dots, \nabla u_n, \mathcal{H}u_1, \dots, \mathcal{H}u_n)\, d\mathbf{x}\,, \tag{2.43}$$

where $\mathbf{x} = (x, y)^\top \in \Omega \subset \mathbb{R}^2$ is a location on a rectangular image plane $\Omega \subset \mathbb{R}^2$ and $\mathbf{u} = (u_1, \dots, u_n)^\top : \Omega \to \mathbb{R}^n$ a vector-valued function. Furthermore, the nabla operator

$$\nabla := (\partial_x, \partial_y) \tag{2.44}$$

is the spatial gradient operator with ∂_* denoting partial derivatives w.r.t. $*$ and

$$\mathcal{H} := \begin{pmatrix} \partial_{xx} & \partial_{xy} \\ \partial_{yx} & \partial_{yy} \end{pmatrix} \tag{2.45}$$

the Hessian operator, such that $\mathcal{H}u_1$ is the Hessian of u_1. Depending on the considered problem and the design choices the integrand F, also called Lagrange-Function, varies. However, independent from the problem and design choices the desired solution is computed as a minimizer of the

energy functional in Equation 2.43. To find such minimizer, the calculus of variations supplies a necessary condition: the so-called Euler-Lagrange equations, which are given by

$$\frac{\partial E}{\partial u_i} = 0 \qquad (i = 1, \dots, n)\,, \tag{2.46}$$

where $\frac{\partial E}{\partial u_i}$ denotes the functional derivatives, which in this case read

$$\begin{aligned}\frac{\partial E}{\partial u_i} = F_{u_i} - \partial_x F_{\partial_x u_i} - \partial_y F_{\partial_y u_i} \\ + \partial_{xx} F_{\partial_{xx} u_i} + \partial_{yx} F_{\partial_{xy} u_i} + \partial_{xy} F_{\partial_{yx} u_i} + \partial_{yy} F_{\partial_{yy} u_i}\,,\end{aligned} \tag{2.47}$$

associated with natural boundary conditions

$$\mathbf{n}^\top \begin{pmatrix} F_{\partial_x u_i} - \partial_x F_{\partial_{xx} u_i} - \partial_y F_{\partial_{xy} u_i} \\ F_{\partial_y u_i} - \partial_x F_{\partial_{yx} u_i} - \partial_y F_{\partial_{yy} u_i} \end{pmatrix} = 0 \qquad (i = 1, \dots, n)\,, \tag{2.48}$$

$$\mathbf{n}^\top \begin{pmatrix} F_{\partial_{xx} u_i} \\ F_{\partial_{xy} u_i} \end{pmatrix} = 0\,, \qquad \mathbf{n}^\top \begin{pmatrix} F_{\partial_{yx} u_i} \\ F_{\partial_{yy} u_i} \end{pmatrix} = 0 \qquad (i = 1, \dots, n)\,, \tag{2.49}$$

where $\mathbf{n}$ is the outer normal vector of the boundary of Ω. A derivation of these boundary conditions is given in [116].

2.4.2 Coarse-to-Fine Warping

Many of the models developed and used throughout this thesis, which reflect the formalized constraints and assumptions, are non-convex energy functionals. Unfortunately, for such non-convex functionals, the calculus of variations introduced in the previous section does not allow to compute a guaranteed global minimizer directly. In particular, it is a highly non-trivial task to minimize such a non-convex energy functional. Hence, a sophisticated minimization strategy is required to obtain a satisfying solution. One well-known and established procedure, especially in the context of optical flow estimation, is the *coarse-to-fine warping approach* of Brox et al. [36]. It relies on an incremental coarse-to-fine fixed point approach which one can interpret as an approximation of the original energy by a series of differential energies. Given the fact that we will extensively use this strategy, we now discuss it in more detail through a simple example for optical flow estimation.

Example Model For our example, we assume that $I_1, I_2 : \Omega \to \mathbb{R}$ are two consecutive image frames of an image sequence and want to estimate the flow field $\mathbf{w} = (u, v)^\top : \Omega \to \mathbb{R}^2$ between the two image frames. To achieve this, we aim at computing the minimizer of an energy functional of the following kind

$$E(\mathbf{w}) = \int_\Omega D(\mathbf{w}) + \alpha \cdot R(\mathbf{w})\, d\mathbf{x}\,, \tag{2.50}$$

where $\mathbf{x} = (x, y)^\top \in \Omega$ denotes the location within the rectangular image domain Ω. More precisely, the energy functional consists of two terms: a data term D and a regularization term R. Furthermore, it contains a weighting parameter α that allows balancing the impact of both terms.

In the case of optical flow estimation, the *data term* D imposes temporal constancy constraints on image features, which in our example is the brightness constancy assumption

$$I_1(\mathbf{x}) = I_2(\mathbf{x} + \mathbf{w}) \,, \tag{2.51}$$

such that corresponding points of a valid solution exhibit the same brightness. Now, we define the data term in such a way that it achieves a minimum value if it fulfills the underlying assumption and a high cost if it violates the assumption. For such settings, a commonly used loss function is the squared difference, which simplifies the minimization process but tends to assign too much weight to outliers. Another common choice is the absolute difference as well as differentiable approximations of such loss, which is less prone to outliers but leads to a more difficult minimization process. In our example case, we stick to the second choice, resulting in

$$D(\mathbf{w}) = \Psi\Big(\big(I_2(\mathbf{x} + \mathbf{w}) - I_1(\mathbf{x})\big)^2\Big) \,, \tag{2.52}$$

where Ψ denotes a regularized linear penalizer given by

$$\Psi(s^2) := \sqrt{s^2 + \epsilon^2} \,, \tag{2.53}$$

with a small $\epsilon > 0$ that ensures differentiability.

The second term in our optical flow example is the *regularization term* R. It is required since we are dealing with an ill-posed problem [27]. For example, one can think of a uniform image region, where considering information at a single location is not sufficient to determine an unambiguous correspondence. To overcome this problem, regularization strategies comprising certain smoothness assumptions can help. They enable the approach to propagate information and dissolve such ambiguities. In the case of optical flow estimation, the most common choice is a first-order smoothness assumption, i.e., the first derivatives vanish,

$$u_x = 0 \,, \quad u_y = 0 \,, \quad v_x = 0 \,, \quad v_y = 0 \,, \tag{2.54}$$

which models a constant flow field. Of course, images sequences typically contain by far more complex motion patterns, but it turns out to be an acceptably good approximation (for regions depicting the same object) and that mainly motion boundaries lead to violations. Similar as for the data term, the resulting regularization term should achieve a minimum value if it fulfills the assumption and a high cost otherwise. As for the data term, a sub-quadratic loss function is advisable, since it allows to capture sharper motion boundaries compared to a quadratic loss function. Hence we use the following regularization term

$$R(\mathbf{w}) = \Psi\big(|\nabla u|^2 + |\nabla v|^2\big) \,. \tag{2.55}$$

Differential Formulation With the example model at hand, we can now turn towards the derivation of the corresponding differential formulation. The first step is to introduce an incremental parametrization, which allows estimating the difference $\mathbf{dw}^k = (du^k, dv^k)^\top$ be-

tween an intermediate solution $\mathbf{w}^k = (u^k, v^k)^\top$ and an updated solution $\mathbf{w}^{k+1} = (u^{k+1}, v^{k+1})^\top$ rather than a final solution directly:

$$\underbrace{\mathbf{w}^{k+1}}_{\text{updated solution}} = \underbrace{\mathbf{w}^k}_{\text{intermediate solution}} + \underbrace{\mathbf{dw}^k}_{\text{unknown increment}} . \tag{2.56}$$

Later on, this parametrization will allow us to introduce an iterative estimation approach based on the concept of warping, to cope with the problem of large displacements. Applying this parametrization to our example model leads to

$$E(\mathbf{dw}^k) = \int_\Omega D(\mathbf{dw}^k) + \alpha \cdot R(\mathbf{dw}^k)\, d\mathbf{x}\,, \tag{2.57}$$

where $D(\mathbf{dw}^k)$ and $R(\mathbf{dw}^k)$ denote the differential counterpart of the original energy formulation, which we specify in the following passages.

The differential counterpart of the data term is obtained by linearizing the original expression w.r.t. the unknown increment $\mathbf{dw}^k$. This linearization not only removes the implicit formulation in the unknowns (in the initial data term $\mathbf{w}$ just appeared as an argument of the image I_2) but also leads to a convex approximation of the original non-convex data term in Equation 2.52. The linearization is performed employing a first-order Taylor expansion that reads

$$I_2(\mathbf{x} + \mathbf{w}^{k+1}) \approx \nabla I_2(\mathbf{x} + \mathbf{w}^k)^\top \mathbf{dw}^k + I_2(\mathbf{x} + \mathbf{w}^k)\,. \tag{2.58}$$

Finally, introducing the abbreviations $I_1 := I_1(\mathbf{x})$, $I_2^k := I_2(\mathbf{x} + \mathbf{w}^k)$ and $I_z^k := I_2^k - I_1$ allow writing the differential formulation of the data term as

$$D(\mathbf{dw}^k) = \Psi\Big(\big(\nabla I_2^{k\top} \mathbf{dw}^k + I_z^k\big)^2\Big)\,. \tag{2.59}$$

In the case of the regularization term, which is already convex, the differential formulation reads

$$R(\mathbf{dw}^k) = \Psi\Big(|\nabla(u^k + du^k)|^2 + |\nabla(v^k + dv^k)|^2\Big)\,. \tag{2.60}$$

Coarse-to-Fine Strategy Another essential ingredient of the minimization approach is the coarse-to-fine strategy, which comes with several advantages. From a mathematical viewpoint, it helps to avoid poor local minima and find a good local or even a global minimum. In terms of the motion estimation problem, it allows us to cope with the large-displacement problem to some extent. Finally, it is computationally less expensive, because the coarser levels are sampled less dense compared to the finest level. Starting from a coarse resolution level $k = 0$, we refine an initial solution $\mathbf{w}^0$ at each fixed point iteration k. To this end, the increment $\mathbf{dw}^k$ is computed, by solving the previously introduced differential formulation, and the new intermediate solution $\mathbf{w}^1$ is evaluated and upsampled to the next finer resolution level. The scale between two successive resolution levels is specified via the downsampling factor $\eta \in (0, 1)$.

But how does this alleviate the aforementioned mentioned problems? The way it helps to avoid local minima can be interpreted as follows. Downsampling the actual problem, i.e., the input frames, to a coarser resolution leads to a smoother energy landscape with less local minima. Hence,

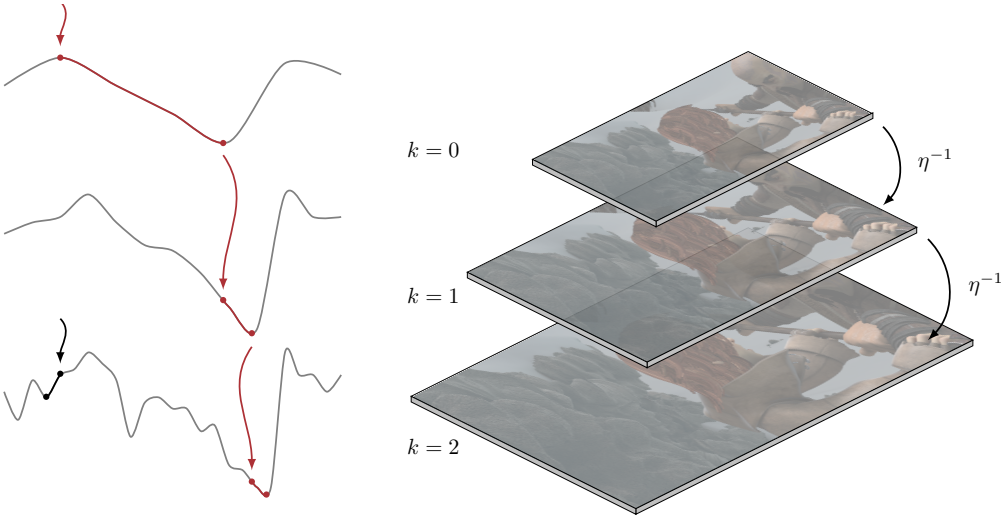


Figure 2.8: Sketch showing how the coarse-to-fine scheme avoids local minima.

it is less likely that the minimization is trapped in a local minima at a coarse resolution. Using the updated solution as initialization, i.e., an intermediate solution, for the next finer resolution level ensures that we are closer to the global optimum or at least a good local minima. Figure 2.8 shows a simple sketch of this in terms of a 2D energy landscape with three levels.

From a problem viewpoint, the coarse-to-fine scheme allows us to cope with the large displacement problem inherited by the approximation via the differential formulation. In particular, due to the performed linearization in the data term, the approximation is only valid for small displacements and does not allow to recover fast motion. However, as a result of downsampling the images to a coarse resolution, large movements are transformed into small displacements, for which the approximation is sufficient. Solely in case of small objects that undergo a large displacement the problem remains since these small objects typically vanish on a coarse resolution. Finally, one must ensure that the movements on finer resolutions remain small. Hence, the images have to be compensated by the motion estimated so far, which takes us to the so-called warping.

WARPING Warping the second image frame I_2 towards the reference frame I_1 by the motion field $\mathbf{w}^k$ can be understood as motion compensation. In our example model, it appears in the data term of the differential formulation, i.e., in the expression $I_2^k := I_2(\mathbf{x} + \mathbf{w}^k)$. The basic idea is to create a new warped image $I_2^{\mathbf{w},k}$ that copies the brightness values of the corresponding locations to the current locations, see Figure 2.9. This reads

$$I_2^{\mathbf{w},k}(\mathbf{x}) := I_2(\mathbf{x} + \mathbf{w}^k)\,. \tag{2.61}$$

In case of a discrete implementation, $\mathbf{x} + \mathbf{w}^k$ will lie, in most cases, between the sampled locations and the actual value must be approximated, e.g., using bilinear interpolation.

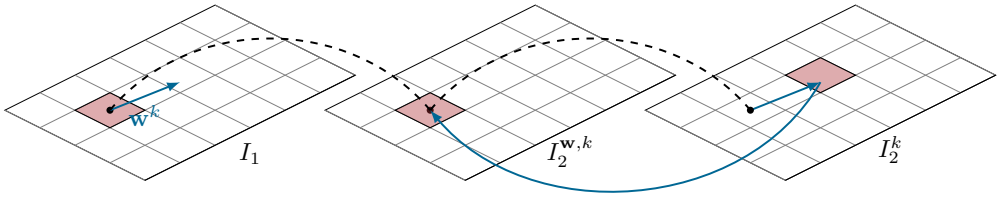


Figure 2.9: Sketch showing the basic implementation of warping.

SOLVING IT To find the minimizer of the differential formulation on every resolution level of the coarse-to-fine fixed-point iteration scheme we make use of the calculus of variations, i.e., we aim at solving the Euler-Lagrange Equations. In our example, they read

$$0 = \Psi'^k_{\text{data}} \cdot \left(\nabla I_2^{k\top} \mathbf{dw}^k + I_z^k\right) \cdot \partial_x I_2^k du^k - \alpha \operatorname{div}\left(\Psi'^k_{\text{reg}} \nabla(u^k + du^k)\right), \tag{2.62}$$

$$0 = \Psi'^k_{\text{data}} \cdot \left(\nabla I_2^{k\top} \mathbf{dw}^k + I_z^k\right) \cdot \partial_y I_2^k dv^k - \alpha \operatorname{div}\left(\Psi'^k_{\text{reg}} \nabla(v^k + dv^k)\right), \tag{2.63}$$

where we used the following abbreviations for the sake of clarity

$$\Psi'^k_{\text{data}} := \Psi'\left(\left(\nabla I_2^{k\top} \mathbf{dw}^k + I_z^k\right)^2\right), \tag{2.64}$$

$$\Psi'^k_{\text{reg}} := \Psi'\left(|\nabla(u^k + du^k)|^2 + |\nabla(v^k + dv^k)|^2\right). \tag{2.65}$$

Furthermore, the boundary conditions are given by

$$\mathbf{n}^\top \nabla du^k = 0, \tag{2.66}$$

$$\mathbf{n}^\top \nabla dv^k = 0. \tag{2.67}$$

Unfortunately, due to the specific choice of the Ψ function in our example, the resulting system of equations is non-linear in the unknown $\mathbf{dw}^k$. To deal with this non-linear system of partial differential equations (PDEs), we apply the Kačanov-type approach [96], which solves the non-linear system through a sequence of linear systems. In particular, we introduce a second fixed point iteration with the iteration index l and keep the non-linear contributions, i.e., Ψ'^k_{data} and Ψ'^k_{reg}, lagging. This modification leads to a system of linear PDEs w.r.t. $\mathbf{dw}^{k,l+1}$ given by

$$0 = \Psi'^{k,l}_{\text{data}} \cdot \left(\nabla I_2^{k\top} \mathbf{dw}^{k,l+1} + I_z^k\right) \cdot \partial_x I_2^k du^{k,l+1} - \alpha \operatorname{div}\left(\Psi'^{k,l}_{\text{reg}} \nabla(u^k + du^{k,l+1})\right), \tag{2.68}$$

$$0 = \Psi'^{k,l}_{\text{data}} \cdot \left(\nabla I_2^{k\top} \mathbf{dw}^{k,l+1} + I_z^k\right) \cdot \partial_y I_2^k dv^{k,l+1} - \alpha \operatorname{div}\left(\Psi'^{k,l}_{\text{reg}} \nabla(v^k + dv^{k,l+1})\right). \tag{2.69}$$

Finally, it takes two steps to solve the system of linear PDEs numerically. In the first step we discretize the system of linear PDEs, e.g., by using standard finite difference approximations in case of the derivatives, and in the second step we apply a method to solve the resulting linear system of equations, e.g., the successive over-relaxation (SOR) method [205].

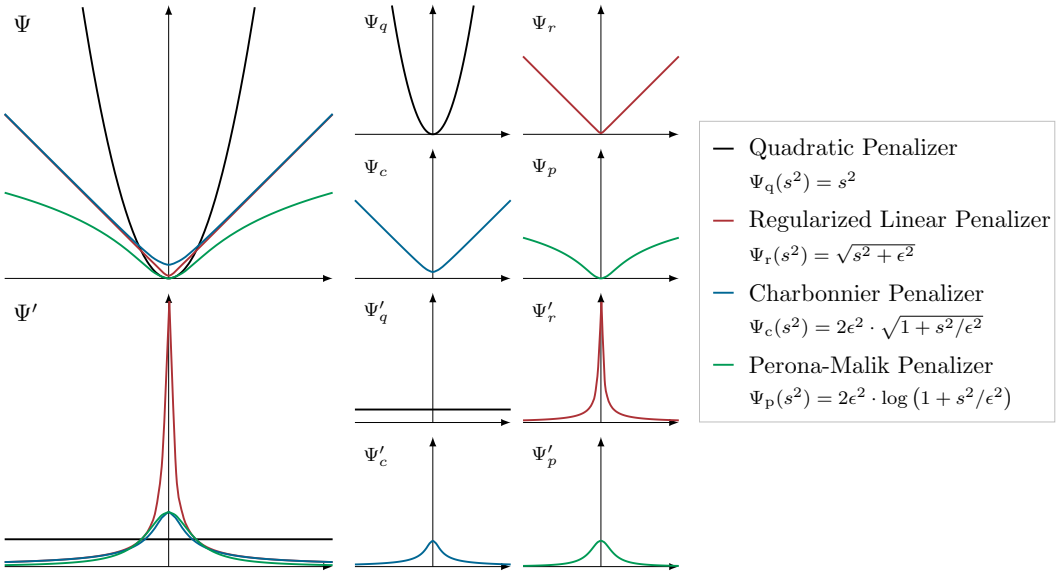


Figure 2.10: Plots of the penalizer functions Ψ (top) as well as the corresponding derivatives Ψ' (bottom).

2.4.3 Modeling Concepts

In the previous section, we introduced the coarse-to-fine warping strategy using an example. This example comprised the derivation of the variational model. In this context, we saw that different design choices arise when formulating the model. Especially, the choice of which penalizer function to use and which type of regularization to employ, recurs multiple times throughout this thesis. Hence, next we detail on these two essential concepts to avoid describing them over and over again.

Penalizer Functions We start with the advantages and disadvantages of different loss functions. Within this thesis, we will use four different differentiable penalizer functions, depicted in Figure 2.10. The first penalizer function is the quadratic penalizer:

$$\Psi_{\mathrm{q}}(s^2) = s^2 \quad \text{with} \quad \Psi'_{\mathrm{q}}(s^2) = 1\,. \tag{2.70}$$

It comes with the nice property of strict convexity as well as the fact that the derivative yields a constant, which simplifies the minimization process. However, due to the quadratic growth possible outliers have a lot of influence. The next two penalizer functions are the regularized linear penalizer of the example model [28]:

$$\Psi_{\mathrm{r}}(s^2) = \sqrt{s^2 + \epsilon^2} \quad \text{with} \quad \Psi'_{\mathrm{r}}(s^2) = \frac{1}{2 \cdot \sqrt{s^2 + \epsilon^2}}\,, \tag{2.71}$$

and the Charbonnier Penalizer [48]:

$$\Psi_{\mathrm{c}}(s^2) = 2\epsilon^2 \cdot \sqrt{1 + s^2/\epsilon^2} \quad \text{with} \quad \Psi'_{\mathrm{c}}(s^2) = \frac{1}{\sqrt{1 + s^2/\epsilon^2}}\,. \tag{2.72}$$

Both penalizer functions are sub-quadratic and therefore yield a more robust behavior in case of outliers. In the context of smoothness terms, such penalizer functions are known to enable an edge-preserving behavior. Furthermore, they are strictly convex, but in contrast to the quadratic penalizer, the derivatives are non-constant, which in case of our example model presented in Section 2.4.2 led to a non-linear system of equations. The fourth penalizer function we will consider is the Perona-Malik Penalizer [134]:

$$\Psi_{\mathrm{p}}(s^2) = 2\epsilon^2 \cdot \log\left(1 + s^2/\epsilon^2\right) \quad \text{with} \quad \Psi'_{\mathrm{p}}(s^2) = \frac{2\epsilon}{s^2 + \epsilon^2}\,. \tag{2.73}$$

It is not only sub-quadratic but also sub-linear, hence it is non-convex. In contrast, to the previous penalizer functions it yields, an edge-enhancing behavior in the context of smoothness terms.

Regularization Another crucial component when dealing with ill-posed problems is regularization. In this thesis, we consider regularization in terms of smoothness assumptions. In this context, one can differentiate between first- and second-order regularization. While the first-order regularization enforces smoothness by assuming the first-order derivatives of the unknowns vanish, second-order regularization imposes regularity by assuming the second-order derivatives vanish. We already used a *first-order regularizer* in our example model for optical flow which reads

$$R_{\text{first-order}}(\mathbf{w}) = \Psi\left(|\nabla u|^2 + |\nabla v|^2\right). \tag{2.74}$$

An exemplary *second-order regularizer* for optical is given by

$$R_{\text{second-order}}(\mathbf{w}) = \Psi\left(|\mathcal{H}u|_F^2 + |\mathcal{H}v|_F^2\right), \tag{2.75}$$

where $\mathcal{H}$ is the Hessian operator and $|\cdot|_F$ is the Frobenius norm. Depending on the considered problem and the chosen parametrization the interpretation of the regularization orders varies.

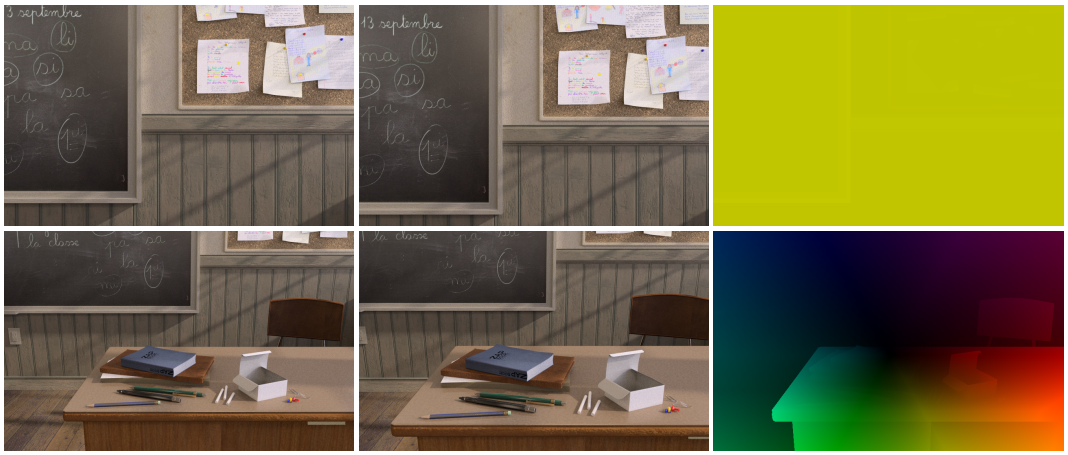


Figure 2.11: *From left to right*: Reference frame, subsequent frame, corresponding motion field. *Top*: Constant motion (first-order smoothness). *Bottom*: Affine motion (second-order smoothness).

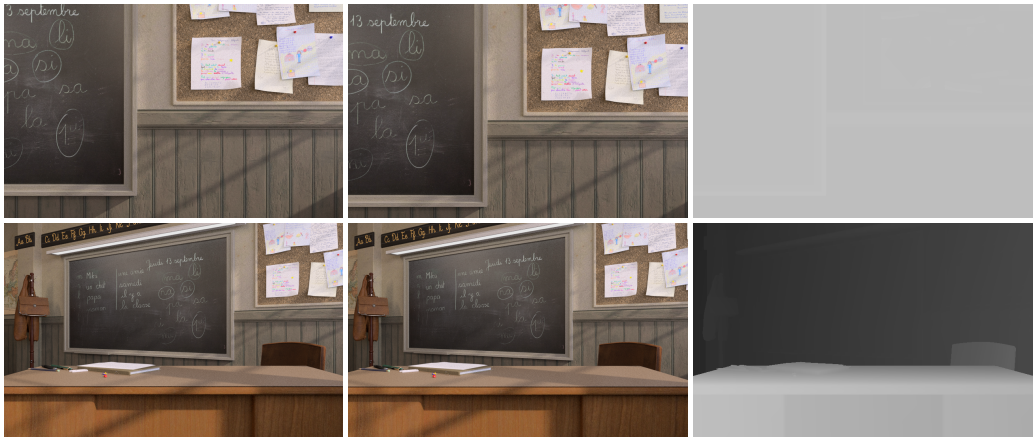


Figure 2.12: *From left to right*: Reference frame, subsequent frame, corresponding depth map. *Top*: Constant depth (first-order smoothness). *Bottom*: Affine depth (second-order smoothness).

For the problem of *motion estimation* first and second-order regularization correspond to different type of movements. Assuming a standard flow parametrization, i.e., a 2D displacement vector per pixel, first-order regularization represents piecewise constant flow fields, which typically occur when planar objects move parallel to the camera (fronto-parallel motion) or vice versa. Figure 2.11 (top) shows an example of such a constant flow field. In the case of second-order regularization piecewise affine motion fields are admissible. Such affine motion patterns occur in scenes where the camera is moving (ego-motion), as shown in Figure 2.11 (bottom).

In the case of *3D reconstruction* first and second-order regularization correspond to different type of shapes. Considering a depth parametrization, i.e., a depth value per pixel, the different regularization orders are related to the previous case. First-order regularization represents piecewise constant depth fields, which occur when planar objects are located parallel to the reference camera. Figure 2.12 (top) displays such a scenario. For the second-order regularization planar objects do not necessary have to lie parallel to the reference camera. This case is shown in Figure 2.12 (bottom).

2.5 Evaluation

After introducing the relevant key aspects of variational modeling, we turn to the last part of the foundation chapter. In this section, we introduce different error measures and visualization techniques that allow us to investigate, evaluate, and compare the performance of our developed methods quantitatively and qualitatively.

2.5.1 Error Measures

Quantitatively benchmarking algorithms has a long tradition and has lead to tremendous progress in the field of computer vision over the last decade. Realizing this procedure takes two ingredients: data for which the correct solution is known (ground truth data) and error measures which allow specifying the performance in terms of numbers.

3D Reconstruction In the case of 3D reconstruction, we consider two commonly used error measures: the root mean square error and the average angular error. The first measure is the *root mean square error* (RMS) against the ground-truth surface/depth-map. By denoting the computed depth as $z : \Omega \to \mathbb{R}$ and the corresponding ground truth as $z_{\text{gt}} : \Omega \to \mathbb{R}$, where Ω is the rectangular image domain, we can formalize it as

$$\text{RMS}(z, z_{\text{gt}}) = \sqrt{\frac{1}{|\Omega|} \int_{\Omega} \big(z(\mathbf{x}) - z_{\text{gt}}(\mathbf{x})\big)^2 \, d\mathbf{x}} \,. \tag{2.76}$$

The second measure we consider is the *average angular error* (AAE) of the surface normals. By denoting the computed normal map as $\mathbf{n} : \Omega \to \mathbb{R}^3$ and the corresponding ground truth as $\mathbf{n}_{\text{gt}} : \Omega \to \mathbb{R}^3$, we can formalize it as

$$\text{AAE}(\mathbf{n}, \mathbf{n}_{\text{gt}}) = \frac{1}{|\Omega|} \int_{\Omega} \arccos\left(\frac{\mathbf{n}(\mathbf{x})^\top \mathbf{n}_{\text{gt}}(\mathbf{x})}{|\mathbf{n}(\mathbf{x})||\mathbf{n}_{\text{gt}}(\mathbf{x})|} \right) d\mathbf{x} \,. \tag{2.77}$$

Motion Estimation For the problem of motion estimation, we also consider two commonly used error measures: the average endpoint error and the bad pixel error. The *average endpoint error* (AEE) describes the average Euclidean difference of two flow fields. By denoting a computed flow field via $\mathbf{w} : \Omega \to \mathbb{R}^2$ and the corresponding ground truth as $\mathbf{w}_{\text{gt}} : \Omega \to \mathbb{R}^2$, we can formalize it as

$$\text{AEE}(\mathbf{w}, \mathbf{w}_{\text{gt}}) = \frac{1}{|\Omega|} \int_{\Omega} \big|\mathbf{w}(\mathbf{x}) - \mathbf{w}_{\text{gt}}(\mathbf{x})\big| \, d\mathbf{x} \,. \tag{2.78}$$

The *bad pixel error* (BP) specifies the percentage of locations for which the endpoint error exceeds a specific threshold τ. The benefit of this metric is that it allows us to consider minor impressions in real-world ground truth data, which my arise in the process of recording and deducing the motion field. Using the same notation as above we can write it as

$$\text{BP}(\mathbf{w}, \mathbf{w}_{\text{gt}}) = \frac{100}{|\Omega|} \int_{\Omega} \chi_\tau(\mathbf{x}) \, d\mathbf{x} \quad \text{with} \quad \chi_\tau(\mathbf{x}) = \begin{cases} 1, & \big|\mathbf{w}(\mathbf{x}) - \mathbf{w}_{\text{gt}}(\mathbf{x})\big| < \tau \\ 0, & \text{else} \end{cases} \tag{2.79}$$

We set $\tau = 3\text{px}$, as it is the default setting of the KITTI 2012 and 2015 benchmarks [65, 119].

2.5.2 Visualizations

While numbers are great for quantitative comparison, they do not allow us to directly assess the quality of the estimation of individual sequences and identify problematic regions. Therefore, we employ different useful visualization techniques.

Depth Visualization To inspect the estimated depth field we make use of two visualization techniques. The first visualization is a color-coding of the depth values, where white denotes close objects and black denotes objects far away. The second visualization is a rendered image, where we use the Lambertian reflectance model and a point light source located in the camera center, to avoid shadows. Figure 2.14 shows an example of both visualizations.

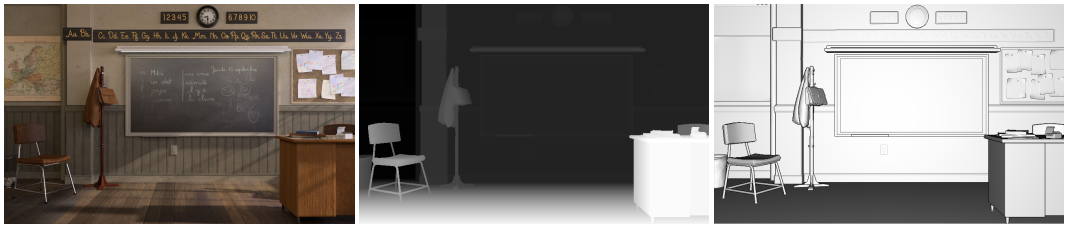


Figure 2.13: *From left to right*: Reference frame, color-coded depth visualization, shaded depth visualization.

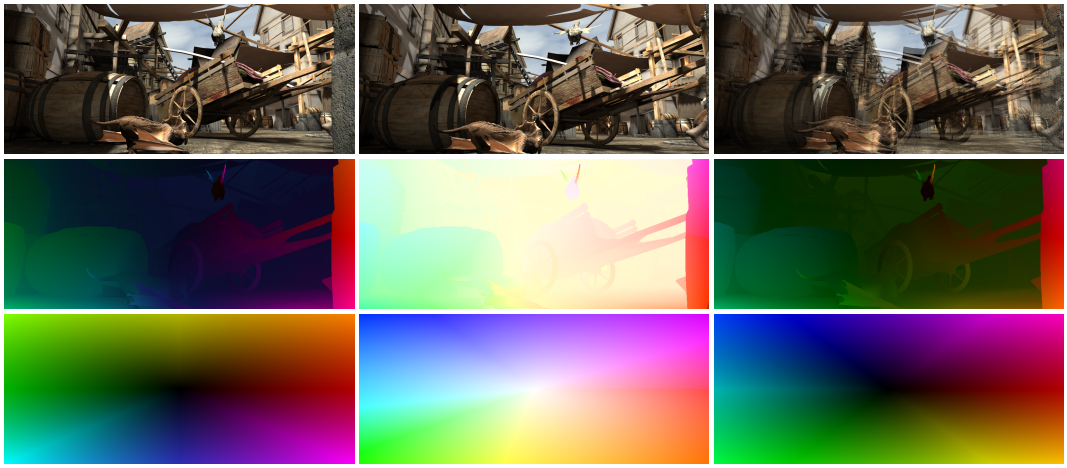


Figure 2.14: Color coding schemes for motion visualization. *First row, from left to right*: Reference frame, second frame, overlayed frames. *Second and third row, from left to right*: Flow fields and corresponding color-scheme of Bruhn [40], Middlebury [23], and KITTI [65].

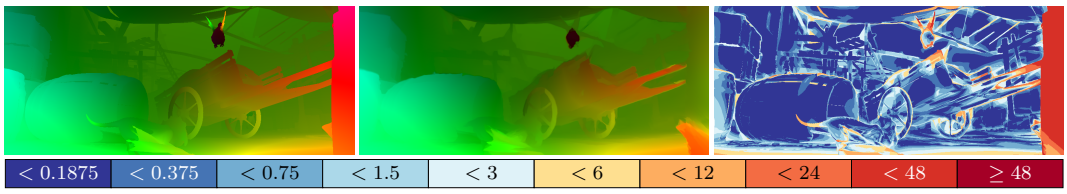


Figure 2.15: *Top, left to right*: Ground truth motion and computed motion, error visualization. *Bottom*: Color representation (numbers denote the endpoint error in terms of pixels).

FLOW VISUALIZATION To visualize flow fields we employ a color-coding of the motion vectors. Within this coding, the color indicates the direction of the displacements and the brightness expresses their magnitude. Figure 2.14 shows an example of three commonly used variants. While this type of visualization enables us to identify sharpness of motion discontinuities, it is hard to rate the pixel-wise accuracy and immediately track down faulty regions. Hence, we also make use of an error visualization. This visualization encodes the per pixel endpoint error. Figure 2.15 shows a ground truth flow field, a computed flow field and the corresponding error visualization.

3 Variational 3D Reconstruction

The task of 3D reconstruction is to capture the shape of real objects. To achieve this goal a variety of different techniques and approaches exist. We can group these techniques into two main categories: active and passive methods. *Active methods* interfere with the actual scene by emitting some sort of light or signals. These methods involve concepts such as structured light, which illuminate the setting with a specially designed light pattern, and time-of-flight, which measures the time-of-flight of an emitted light signal between the camera and multiple object points.

In contrast, *passive methods* do not directly interfere with the scene and only capture a single or multiple images. We can further group the passive methods by the type of depth cue utilized to recover the shape. Monocular cues, for example, require only a single image and exploit information such as shading, texture or silhouettes. Binocular cues need multiple images captured from different viewpoints and allow to use the so-called parallax, the displacement in the apparent position.

In this chapter, we present a variational approach that simultaneously exploits two fundamentally different depth cues for passive 3D reconstruction, i.e., the shading cue and the parallax cue. Main parts of this chapter are based on the work published in [3, 4, 5].

3.1 Introduction

Approaches that exploit parallax cues to reconstruct a 3D surface are known as *stereo methods*. By identifying corresponding pixels in multiple images and triangulating them, stereo methods can recover the actual 3D shape. On the other hand, approaches that exploit shading cues for 3D reconstruction are known as *shape from shading* (SfS) methods. Using a reflection model that relates the image brightness, i.e., the shading, to the surface normal allows to recover the shape.

Both techniques are quite complementary. Stereo methods benefit from highly textured regions since they support the process of finding correspondences. However, at the same time, such methods also require sophisticated regularization strategies to deal with ambiguities, possibly emerging from weakly-textured and homogeneous regions, which may lead to over-smooth results and less detailed reconstructions. In contrast, SfS benefits from un-textured homogeneous objects, since only in this case observed brightness changes could be directly attributed to depth changes rather than ending up in an ambiguity between a color and a depth change. While in such an ideal scenario without texture, only a little or no regularization is required, it is typically still needed to cope with ambiguities arising in textured regions.

Knowing these advantages and drawbacks of both strategies, it appears quite natural to fuse stereo and SfS techniques to improve the reconstruction accuracy. Since the first ideas of Blake et al. [29] in 1985, researchers have proposed a variety of methods for combining stereo and SfS.

3.1.1 Related Work

We can divide the literature of 3D reconstruction methods that combine both parallax and shading cues into three groups: fusion approaches, sequential approaches, and joint approaches. In the following, we review all three groups.

Fusion Approaches The first group comprises fusion approaches. Such approaches perform stereo and SfS independently of each other and combine the results in terms of a sophisticated post-processing step. Examples are, for instance, the method of Cryer et al. [52] that fuses depth maps from stereo and SfS in the frequency domain or the approach of Haines and Wilson [77] that combines disparity information and surface normals within a probabilistic approach. Since the initial computations are performed separately, a direct interaction between the cues is not possible. Although fusing the information may allow improving the results, the quality gain is typically somewhat limited compared to more integrated strategies.

Sequential Approaches In contrast to fusion approaches, sequential techniques perform the stereo and SfS computation consecutively, where stereo provides an initialization for SfS. Consequently, one may consider these techniques as shading-based refinement methods. First approaches such as the method of Leclerc and Bobick [106] and Hougen and Ahuja [83] have been restricted to a simple orthographic camera model and a constant albedo, while assuming a global light direction and a polynomially parametrized reflectance map, respectively. Following the work of Fua and Leclerc [60], we denote these techniques as *view-centered*, since they perform the refinement in the pixel domain. In contrast, so-called *object-centered* approaches operate directly on a complete surface representation, e.g., an initial closed 3-D mesh of an object. While they typically rely on a preceding stereo approach to obtain an initial solution, parallax cues are only implicitly exploited during the refinement by imposing shading cues on multiple views. As in the view-centered case, most of the object-centered approaches assume that the scene consists of a single material [194, 197, 208, 209]. They either focus on generalizing the reflectance model for dealing with non-Lambertian surfaces, e.g., by using the Phong model [208] or a general parametrization in terms of a view-independent reflectance map [209], or they aim towards estimating the illumination, e.g., by using spherical harmonics [194] or a general illumination vector field [197]. Among the few exceptions that do not rely on the single material assumption are the approach of Yoon et al. [204] that estimates the reflectance of a dichromatic surface for a given illumination, and the approach of Valgaerts et al. [171] that exploits temporal constraints on clustering a spatially varying albedo in the context of facial performance capture.

Instead of using stereo information, there are also sequential approaches [78, 130, 195, 207, 218] that make use of depth measurements obtained via active reconstruction methods, e.g., RGB-D cameras. Although this information is typically rather noisy, the provided depth and the corresponding surface normals simplify the estimation of global illumination parameters, e.g., the coefficients of spherical harmonics, and albedo maps significantly compared to stereo-based methods. Again, approaches range from methods that assume a uniform albedo [78] to strategies that cluster different albedo regions [207]. Recently, researchers also proposed RGB-D based techniques that operate in real-time [130, 195]. In general, however, such methods need dedicated hardware, e.g., time-of-flight cameras, for the active reconstruction.

As expected, in the case of sequential approaches, the shading-based refinement can benefit significantly from the preceding stereo reconstruction. However, there is no direct feedback in the

sense that stereo cannot take advantage of any shading cues. Strictly speaking, this holds for methods that involve active components as well, since they typically do not include shading cues at all. Nevertheless, such methods clearly show that, when having a reasonable initial depth, estimating the illumination and albedo jointly seems to be very beneficial in terms of reconstruction quality.

Joint Approaches In contrast to sequential and fusion methods, joint approaches exploit parallax and shading cues simultaneously when estimating the depth. For example, Fua and Leclerc [60] proposed to minimize an objective function with stereo, shading and smoothness terms, that allows a slowly varying albedo but requires a known illumination. Moreover, in the context of face reconstruction Samaras et al. [148] developed a method that fits a face model to the stereo data and refines it while re-estimating illumination and albedo. As most of the joint approaches, these methods rely on a preceding stereo estimation. This pre-estimation is required to obtain a non-trivial initialization of the underlying surface parametrization, e.g., an initial mesh or a volumetric signed distance function model. More recently, Langguth et al. [105] proposed a joint approach that combines stereo and shading cues within a combined energy. While this approach builds on the Retinex assumption and is hence able to estimate the depth almost independently of the albedo, it relies on a preceding estimation of the illumination from an initial stereo result.

A method that does not require such an initial mesh as the previous techniques is the level set approach of Jin et al. [93]. However, although the corresponding model considers ambient light as well as an explicit background, it is restricted to two regions with constant albedo as well as to a global light direction. Moreover, by relying on multi-view SfS instead of multi-view stereo, parallax cues are only exploited implicitly, i.e., different views are compared to the correspondingly rendered images of the reconstruction (image-to-model, SfS), but no direct matching between the input images is performed (image-to-image, stereo).

In face of the existing literature, it would evidently be desirable to develop a joint approach that simultaneously exploits parallax and shading cues to estimate depth, illumination, and albedo from scratch, i.e., a general method that does not need an initial estimate.

3.1.2 Contributions

In this chapter, we present such a joint approach. We propose a novel view-centered method that combines data terms from stereo and SfS based on a separate parametrization for depth, illumination, and albedo. In the reconstruction process, the parallax cue allows a robust estimation of the object surface, while the shading cue enables the recovery of fine surface details. In this context, the careful selection of the regularization plays an important role. While we make use of specially tailored anisotropic first-order smoothness terms that provide sharp illumination and albedo maps, we employ an anisotropic second-order smoothness term for the depth that allows reconstruction of slanted surfaces. As a result, we obtain a method that enables the estimation of high-quality depth maps of Lambertian scenes with varying albedo under unknown illumination.

However, our contributions are not limited to the modeling side only. Also from a numerical viewpoint, we provide some novel ideas. In particular, we propose a coarse-to-fine minimization scheme based on a linearization of all data terms. This scheme does not only allow us to estimate all unknowns simultaneously but also to embed the entire optimization into a hierarchical incremental fixed point strategy, as described in Subsection 2.4.2. Due to the use of upwind schemes for approximating the derivatives in the SfS data term, we term this strategy *hyperbolic warping*.

3.2 Variational Model

In this section, we propose our variational model which allows exploiting shading and parallax cues jointly. To this end, we first introduce the considered setting together with the utilized parametrization and then focus on the actual model, including a detailed discussion of all terms.

3.2.1 Setting and Parametrization

Our setting consists of n perspective cameras C_i ($i \in \{1, ..., n-1\}, n \geq 2$), which capture the scene from varying viewpoints in terms of RGB color images $\mathbf{I}_i : \Omega_i \to \mathbb{R}^3$. In particular, we assume the cameras to be calibrated, i.e., the corresponding camera projection matrices

$$P_i = K_i[R_i|\mathbf{t}_i]\,, \tag{3.1}$$

as described in Subsection 2.2.2, are known a priori. Furthermore, we assume w.l.o.g. that the reference camera C_0 is aligned with the world coordinate frame. Therefore, we parametrize the unknown surface of the corresponding back projection, see Subsection 2.2.3, as

$$\mathbf{s}(\mathbf{x}, z) = z(\mathbf{x}) \cdot K_0^{-1}\tilde{\mathbf{x}}\,, \tag{3.2}$$

where $z(\mathbf{x})$ is the depth measured along the optical axis, K_0 is the camera calibration matrix of the reference camera and $\tilde{\mathbf{x}} = (x, y, 1)^\top$ denotes the homogeneous counterpart of the location $\mathbf{x} = (x, y)^\top \in \Omega$ in the reference frame, see Figure 3.1.

Moreover, to exploit shading cues in a fairly unconstrained setting, we not only take the surface depth into account but also consider the scene illumination and the surface reflectance properties. To derive a suitable parametrization for the latter two, we contemplate the image formation process in terms of the rendering equation, as introduced in Subsection 2.3.3. Besides, we assume the surface is Lambertian and does not emit any radiance. Hence, the rendering equation simplifies to

$$L_o(\mathbf{s}, \omega_o) = \int_{\Omega_\mathbf{s}} \frac{\rho}{\pi} L_i(\mathbf{s}, \omega_i)(\omega_i \cdot \mathbf{n})\, d\omega_i\,, \tag{3.3}$$

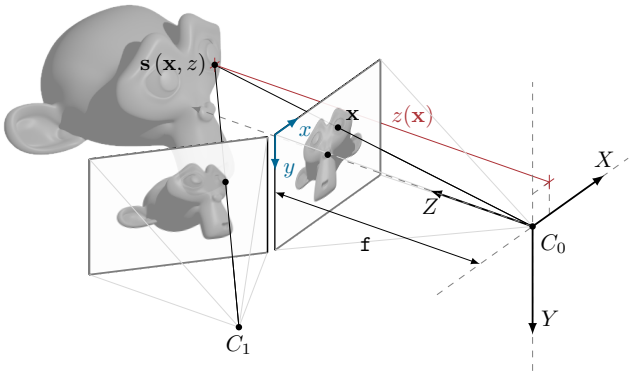


Figure 3.1: Surface parametrization via the back projection $\mathbf{s}(\mathbf{x}, z)$.

where $\Omega_{\mathbf{s}}$ represents the upper unit hemisphere centered around the surface normal $\mathbf{n}$ at the corresponding surface point $\mathbf{s} = \mathbf{s}(\mathbf{x}, z)$, and $L_i(\mathbf{s}, \omega_i)$ stands for the incident radiance from direction ω_i at $\mathbf{s}$. Finally, we assume that the reference camera captures the reflected radiance of a specific wavelength spectrum in terms of the corresponding color image $\mathbf{I}_0 = (I_0^1, I_0^2, I_0^3)^\top$, such that

$$\mathbf{I}_0(\mathbf{x}) = \int_{\Omega_{\mathbf{s}}} \boldsymbol{\rho}\, L(\mathbf{s}, \omega_i)(\omega_i \cdot \mathbf{n})\, d\omega_i \,, \tag{3.4}$$

where $\boldsymbol{\rho} = (\rho_R, \rho_G, \rho_B)^\top : \Omega \to \mathbb{R}^3$ is considered a vector-valued albedo that represents the reflectivity for the individual wavelength spectra. By separating the albedo $\boldsymbol{\rho}$ and the surface normal $\mathbf{n}$ from the integrand, the integral only contains the contributions of the incident radiance, which we summarize in terms of an illumination vector field $\mathbf{l} = (l_0, l_1, l_2)^\top : \Omega \to \mathbb{R}^3$ (see Figure 3.2)

$$\mathbf{I}_0(\mathbf{x}) = \boldsymbol{\rho}(\mathbf{x}) \underbrace{\left(\int_{\Omega_{\mathbf{s}}} L(\mathbf{s}, \omega_i)\, \omega_i\, d\omega_i \right)^\top}_{=\mathbf{l}(\mathbf{x})} \mathbf{n}(\mathbf{x}) \,. \tag{3.5}$$

Xu et al. [197] and Queau et al. [135] consider similar parametrizations that also include an illumination vector. In contrast to the work of Xu et al., that assumes the albedo to be constant, we explicitly separate it from the illumination vector field. This choice enables the application to more realistic scenarios where the albedo is spatially varying. In contrast to the work of Queau et al., that uses a single channel variant operating on grayscale images in the context of photometric stereo [135], we employ a multi channel variant. This choice allows us to deal with RGB color images.

In a nutshell, three functions form the final parametrization: the depth map z, the illumination vector field $\mathbf{l}$, and the vector-valued albedo map $\boldsymbol{\rho}$. This parametrization enables the application to fairly unconstrained settings without the need for a tedious illumination calibration.

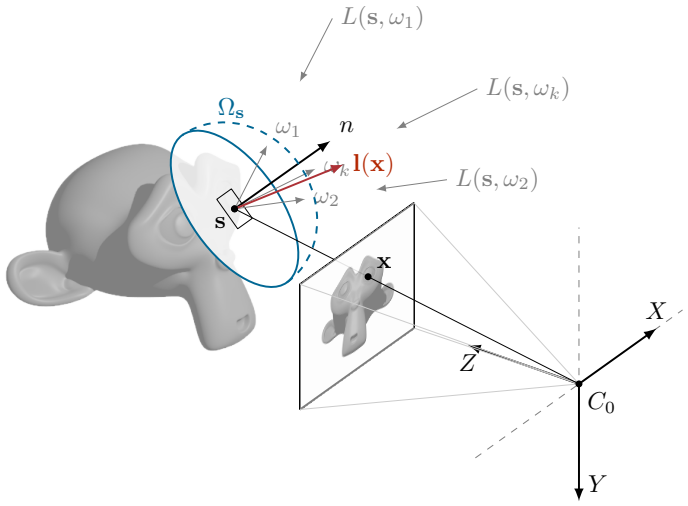


Figure 3.2: Illumination parametrization via the illumination vector $\mathbf{l}(\mathbf{x})$.

3.2.2 Variational Model

Using the previous parametrization, we propose to compute all three unknowns, namely the depth z, the illumination vector $\mathbf{l}$, and the albedo $\boldsymbol{\rho}$, as a minimizer of the following energy functional:

$$E(z, \mathbf{l}, \boldsymbol{\rho}) = D_{\text{stereo}}(z) + \nu \cdot D_{\text{sfs}}(z, \mathbf{l}, \boldsymbol{\rho}) + \alpha_z \cdot R_{\text{depth}}(z) + \alpha_{\mathbf{l}} \cdot R_{\text{illum}}(\mathbf{l}) + \alpha_{\boldsymbol{\rho}} \cdot R_{\text{albedo}}(\boldsymbol{\rho}) \,. \quad (3.6)$$

It is composed of two data terms and three regularization terms. While the stereo data term D_{stereo} exploits parallax cues by accounting for a photo-consistency between the reference image and the other match images, the shape from shading data term D_{sfs} exploits shading cues by relating the reference image and a rendered image based on depth, illumination, and albedo. To resolve ambiguities between the unknowns, the three regularizers R_{depth}, R_{illum}, and R_{albedo} have been added. Finally, the positive weights ν, α_z, $\alpha_{\mathbf{l}}$, and $\alpha_{\boldsymbol{\rho}}$ balance the terms. Let us now discuss the different data and smoothness terms in detail.

Stereo Data Term For the multi-view stereo data term we consider the depth-parametrized model of Robert and Deriche [145] based on the photo-consistency, i.e., the brightness constancy, of projected surface points. Recent stereo approaches frequently use this model, see e.g., [26, 127, 155]. To account for slight illumination changes between subsequently recorded views, we complement it by a gradient constancy assumption [36, 155]. As shown in the work of Semerjian [155] this assumption can be interpreted as a patch similarity with infinitesimal patches. Hence, we obtain

$$D_{\text{stereo}}(z) = \frac{1}{n-1} \sum_{i=1}^{n-1} \int_{\Omega} \Psi_r\Big(|\mathbf{I}_0(\mathbf{x}) - \mathbf{I}_i(\mathbf{x}_i)|^2\Big) + \gamma \cdot \Psi_r\Big(|\mathcal{J}(\mathbf{I}_0(\mathbf{x})) - \mathcal{J}(\mathbf{I}_i(\mathbf{x}_i))|_F^2\Big) \, \mathrm{d}\mathbf{x} \,, \quad (3.7)$$

where $\gamma = 1$ is a positive weight to balance both constancy assumptions, $\mathcal{J}(\mathbf{I}_0(\mathbf{x}))$ and $\mathcal{J}(\mathbf{I}_i(\mathbf{x}_i))$ are the Jacobians of $\mathbf{I}_0(\mathbf{x})$ and $\mathbf{I}_i(\mathbf{x}_i)$, respectively, that contain the partial derivatives w.r.t. x and y, $|\cdot|_F$ denotes the Frobenius norm, and

$$\mathbf{x}_i = \pi(\mathbf{s}(\mathbf{x}, z)) = \pi(P_i \, \tilde{\mathbf{s}}(\mathbf{x}, z)) \quad (3.8)$$

is the projection of the surface point $\mathbf{s}$ at the reference location $\mathbf{x}$ onto the image $\mathbf{I}_i$, see Figure 3.3.

Finally, to improve the robustness of both assumptions w.r.t. outliers and occlusions, we model the data term more robust by applying the regularized linear norm. To this end, we follow Bruhn and Weickert [38] and penalize both constancy assumptions separately.

Shape from Shading Data Term To model the SfS data term, we make use of the previously stated rendering equation with the compact illumination vector field parametrization, see Equation 3.5. Introducing the following reflectance function

$$\mathbf{R}(\mathbf{x}) = \boldsymbol{\rho}(\mathbf{x})(\mathbf{l}(\mathbf{x})^\top \mathbf{n}(\mathbf{x})) \,, \quad (3.9)$$

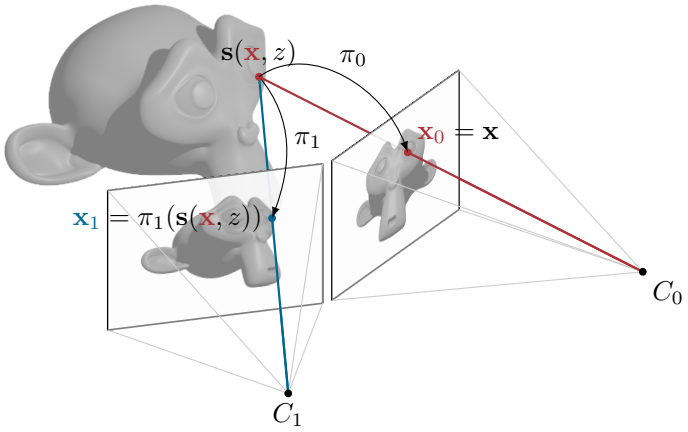


Figure 3.3: Illustration of the stereo geometry. While the pixel location $\mathbf{x}_i$ depends on the object surface $\mathbf{s}(\mathbf{x}, z)$ via the projection $\pi_i(\mathbf{s}(\mathbf{x}))$, the surface $\mathbf{s}$ itself depends on $\mathbf{x}$ via the depth $z(\mathbf{x})$.

that evaluates the rendering equation given a unit surface normal $\mathbf{n}$, the illumination vector field $\mathbf{l}$ and the vector-valued albedo $\boldsymbol{\rho}$, we can write the SfS data term as

$$D_{\text{sfs}}(z, \mathbf{l}, \boldsymbol{\rho}) = \int_\Omega |\mathbf{I}_0(\mathbf{x}) - \mathbf{R}(\mathbf{x})|^2 \, \mathrm{d}\mathbf{x} \,, \tag{3.10}$$

which relates the reference image $\mathbf{I}_0$ to the introduced reflectance function. Please note that the depth estimates appear in the reflectance function $\mathbf{R}$ in terms of the surface normal $\mathbf{n}$. To establish the connection between the surface normal $\mathbf{n}$ and the depth z, we refer to derivation given in Subsection 2.2.3, which results in

$$\mathbf{n}(\mathbf{x}) = -\frac{\bar{\mathbf{n}}(\mathbf{x})}{|\bar{\mathbf{n}}(\mathbf{x})|} \quad \text{with} \quad \bar{\mathbf{n}}(\mathbf{x}) = \begin{pmatrix} -s_x z_x \\ -s_y z_y \\ \nabla z^\top (\mathbf{x} - \mathbf{o}) + z \end{pmatrix} . \tag{3.11}$$

Depth Regularization To allow for a smooth reconstruction of slanted surfaces, we refrain from using a first-order regularizer that inherently favors fronto-parallel surfaces. Instead, we resort to a second-order regularization strategy that enables the model to recover linear depth changes [1,138,151,177]. In particular, we employ the anisotropic second-order regularizer of Hafner et al. [75] that originated in the context of focus fusion. It combines the edge preservation properties of a second-order coupling model, e.g., the total generalized variation (TGV) [34], with a direction-dependent adaptation behavior, which allows to consider the underlying image information to guide the regularization. The corresponding regularizer reads

$$R_{\text{depth}}(z) = \inf_{\mathbf{a}} \int_\Omega C(z, \mathbf{a}) + \alpha_{\mathbf{a}} \cdot S(\mathbf{a}) \, \mathrm{d}\mathbf{x} \,, \tag{3.12}$$

and consists of two terms: the coupling term $C(z, \mathbf{a})$ and the smoothness term $S(\mathbf{a})$:

$$C(z, \mathbf{a}) = \sum_{l=1}^{2} \Psi_l \left(\left(\mathbf{r}_l^\top (\nabla z - \mathbf{a}) \right)^2 \right), \tag{3.13}$$

$$S(\mathbf{a}) = \sum_{l=1}^{2} \Psi_l \left(\sum_{m=1}^{2} \left(\mathbf{r}_m^\top \mathcal{J}(\mathbf{a})\, \mathbf{r}_l \right)^2 \right), \tag{3.14}$$

where $\mathbf{a} : \Omega \to \mathbb{R}^2$ is a vector-valued auxiliary function, $\mathcal{J}(\mathbf{a})$ is the Jacobian of $\mathbf{a}$, and $\mathbf{r}_1$ and $\mathbf{r}_2$ denote orthogonal unit vectors that correspond to the dominant directions of the local structure of the reference image $\mathbf{I}_0 = \left(I_0^1, I_0^2, I_0^3\right)^\top$, respectively. In our model, we compute the directions $\mathbf{r}_1$ and $\mathbf{r}_2$ as eigenvectors of the color structure tensor [56, 59]

$$J := K_{\sigma_o} * \sum_{c=1}^{3} \left(\nabla (K_{\sigma_i} * I_0^c)\, \nabla (K_{\sigma_i} * I_0^c)^\top \right), \tag{3.15}$$

where K_{σ_i} and K_{σ_o} are spatial Gaussians with standard deviation σ_i and σ_o for pre-smoothing and local integration, respectively, and $*$ is the convolution operator.

Let us now detail on the two terms that are balanced by the parameter $\alpha_{\mathbf{a}}$. The coupling term connects the gradient of the depth map ∇z to the auxiliary function $\mathbf{a}$. Hence, $\mathbf{a}$ can be considered an approximation of the first-order depth map derivatives. The smoothness term ensures that the Jacobian $\mathcal{J}(\mathbf{a})$ of the auxiliary function is small. Consequently, it enforces a first-order smoothness constraint on the auxiliary function $\mathbf{a}$. Both terms together realize a second-order regularization on z. In fact, for the particular case that the coupling term is perfectly fulfilled, i.e., $\mathbf{a} = \nabla z$, the smoothness term penalizes the second-order directional derivatives $z_{\mathbf{r}_1\mathbf{r}_1}$, $z_{\mathbf{r}_1\mathbf{r}_2}$, $z_{\mathbf{r}_2\mathbf{r}_1}$ and $z_{\mathbf{r}_2\mathbf{r}_2}$. In this case, the resulting regularizer comes down to an anisotropic direct second-order variant.

So far, we have discussed how to realize the second-order regularization by using the auxiliary function $\mathbf{a}$. Let us now explain how to achieve the desired anisotropic behavior. The central concept in this context is the separate sub-quadratic penalization of the two directions $\mathbf{r}_1$ and $\mathbf{r}_2$ in the coupling term and the smoothness term, respectively. This separate penalization not only adapts the regularization to the local image structure by considering the directions $\mathbf{r}_1$ and $\mathbf{r}_2$ from the structure tensor J, but also allows to preserve edges in both directions independently. This behavior, in turn, allows coping with different structural scenarios such as corners, edges and uniform areas. Furthermore, applying the separate penalization to the smoothness term and the coupling term has different effects. While in case of the smoothness term it yields an anisotropic regularization of the auxiliary function $\mathbf{a}$, it leads to an anisotropic regularization of the depth z in case of the coupling term. In the latter case, we penalize only those deviations from ∇z that the piecewise smooth auxiliary function $\mathbf{a}$ cannot explain. Such a function $\mathbf{a}$, in turn, corresponds to a piecewise affine depth z which makes once again the second-order regularization explicit.

As penalizing functions for the coupling and the smoothness terms we chose the edge-enhancing Perona-Malik penalizer $\Psi_1 = \Psi_p$ along the dominant $\mathbf{r}_1$-direction with $\epsilon = 0.001$, and the edge-preserving Charbonnier function $\Psi_2 = \Psi_c$ orthogonal to it with $\epsilon = 0.01$, i.e., in $\mathbf{r}_2$-direction, as proposed in the work of Volz et al. [179].

Illumination Regularization As shown by Xu et al. [197], the illumination vector $\mathbf{l}$ is typically piecewise constant or only varies smoothly across the surface. Hence, a first-order regularization strategy is an appropriate choice. In particular, we make use of an anisotropic first-order smoothness term that exploits directional information and thus allows to capture more details. The corresponding smoothness term is given by

$$R_{\text{illum}}(\mathbf{l}) = \int_{\Omega} \sum_{l=1}^{2} \Psi_l\left(\left|\mathcal{J}(\mathbf{l})\,\mathbf{r}_l\right|^2\right) \mathrm{d}\mathbf{x}\,, \tag{3.16}$$

where $\mathcal{J}(\mathbf{l})$ denotes the Jacobian of $\mathbf{l}$. As in the case of the depth regularization, the penalizer functions Ψ_1, Ψ_2 are chosen to be the Perona-Malik and Charbonnier penalizer with $\epsilon = 0.01$, respectively, and the directions $\mathbf{r}_1$, $\mathbf{r}_2$ are obtained as the eigenvectors of the color structure tensor J of the reference image, see Equation 3.15.

Albedo Regularization Modeling the regularization term for the albedo, we rely on a common assumption from the field of intrinsic image decomposition. There, it has been observed that pixels with similar chromaticity are likely to share a similar albedo [49]. Since we are interested in separating albedo from geometry and illumination, we follow this idea and make use of a first-order smoothness term which reduces smoothness at chromaticity edges. This behavior is achieved using a positive, decreasing weighting function g applied to the directional derivatives of the rg-chromaticity, which serves as a fuzzy edge detector for chromaticity edges. Hence, we propose the following anisotropic smoothness term that allows the preservation of fine structures in the albedo

$$R_{\text{albedo}}(\boldsymbol{\rho}) = \int_{\Omega} \sum_{l=1}^{2} g\left(\left|\mathcal{J}(\mathbf{ch}(\mathbf{I_0}))\,\mathbf{r}_l\right|^2\right) \cdot \Psi_l\left(\left|\mathcal{J}(\boldsymbol{\rho})\,\mathbf{r}_l\right|^2\right) \mathrm{d}\mathbf{x}\,, \tag{3.17}$$

where $\mathcal{J}(\mathbf{ch}(\mathbf{I_0}))$ and $\mathcal{J}(\boldsymbol{\rho})$ are the Jacobians of the chromaticity and albedo, respectively, and

$$\mathbf{ch}(\mathbf{I_0}) = \frac{\mathbf{I_0}}{I_0^1 + I_0^2 + I_0^3} \tag{3.18}$$

denotes the rg-chromaticity, obtained by a pixel-wise normalization of the RGB values. In this context, we set $g(s^2) = \Psi_p' = 1/(1+s^2/\epsilon^2)$ to be the Perona-Malik diffusivity. This choice leads to the fact that jumps in the albedo map mainly align with chromaticity edges since $g(s^2) \approx 0$ for large arguments $s^2 \gg \epsilon^2$. As before, we choose the penalizer functions Ψ_1, Ψ_2 to be the Perona-Malik and Charbonnier penalizer with $\epsilon = 0.01$, respectively.

3.3 Minimization

To compute the minimizer of the energy functional given in Equation 3.6, we build upon the coarse-to-fine warping strategy introduced in Subsection 2.4.2. Hence, we first derive the differential formulation of the energy functional that we have to solve at each resolution level of the coarse-to-fine scheme. Furthermore, we explain the numerical solution as well as relevant implementation details. In the course of this we also provide the associated Euler-Lagrange equations.

3.3.1 Differential Formulation

Let us start by introducing the incremental formulation of all unknowns, given by

$$z^{k+1} = z^k + dz^k\,, \tag{3.19}$$
$$\mathbf{l}^{k+1} = \mathbf{l}^k + \mathbf{dl}^k\,, \tag{3.20}$$
$$\boldsymbol{\rho}^{k+1} = \boldsymbol{\rho}^k + \mathbf{d}\boldsymbol{\rho}^k\,, \tag{3.21}$$

where z^k, $\mathbf{l}^k$, and $\boldsymbol{\rho}^k$ denote the known intermediate solutions and dz^k, $\mathbf{dl}^k$, and $\mathbf{d}\boldsymbol{\rho}^k$ are the unknown increments at the resolution level k within the coarse-to-fine pyramid. The differential energy w.r.t. the unknown increments is then given by

$$\begin{aligned} E^k\big(dz^k, \mathbf{dl}^k, \mathbf{d}\boldsymbol{\rho}^k\big) = D^k_{\text{stereo}}\big(dz^k\big) + \nu \cdot D^k_{\text{sfs}}\big(dz^k, \mathbf{dl}^k, \mathbf{d}\boldsymbol{\rho}^k\big) + \alpha_z \cdot R^k_{\text{depth}}\big(dz^k\big) \\ + \alpha_{\mathbf{l}} \cdot R^k_{\text{illum}}\big(\mathbf{dl}^k\big) + \alpha_{\boldsymbol{\rho}} \cdot R^k_{\text{albedo}}\big(\mathbf{d}\boldsymbol{\rho}^k\big) + \alpha_{\text{inc}} \cdot R^k_{\text{inc}}\big(dz^k, \mathbf{dl^k}, \mathbf{d}\boldsymbol{\rho^k}\big)\,. \end{aligned} \tag{3.22}$$

Please note that, compared to the original energy, we introduced an additional term R^k_{inc}, which ensures that increments are sufficiently small. We explain the term later on. After we have outlined the basic structure of the differential energy, let us now discuss the different terms in detail.

Differential Stereo Data Term We obtain the differential formulation of the stereo data term by linearizing the original formulation in Equation 3.7 w.r.t. the depth increment dz^k. To this end, let us introduce the following three abbreviations

$$\varphi_i^{k,c}(\mathbf{x}) := I_0^c(\mathbf{x}) - I_i^c(\mathbf{x_i^k})\,, \tag{3.23}$$
$$\varphi_{i,x}^{k,c}(\mathbf{x}) := \partial_x\, I_0^c(\mathbf{x}) - \partial_x\, I_i^c(\mathbf{x}_i^k)\,, \quad \text{and} \quad \varphi_{i,y}^{k,c}(\mathbf{x}) := \partial_y\, I_0^c(\mathbf{x}) - \partial_y\, I_i^c(\mathbf{x}_i^k)\,, \tag{3.24}$$

for the brightness and gradient constancy assumption per color channel $c \in \{1, 2, 3\}$, where $\mathbf{x}_i^k = \pi_i^k(\mathbf{s}(\mathbf{x}, z^k))$ denotes the projection of the surface point $\mathbf{s}(\mathbf{x}, z^k)$, corresponding to location $\mathbf{x}$ with the current depth estimate z^k, onto the image $\mathbf{I}_i$. Here π_i^k denotes the projection performed of the i-th camera with the corresponding projection matrix scaled to match the respective resolution level k of the coarse-to-fine pyramid. This allows us to state the linearized constraints for the brightness and gradient constancy assumption, respectively, as follows:

$$\bar{\varphi}_{i,0}^{k,c} := \varphi_i^{k,c}(\mathbf{x}) + \frac{\partial \varphi_i^{k,c}(\mathbf{x})}{\partial z^k(\mathbf{x})} \cdot dz^k(\mathbf{x})\,, \tag{3.25}$$
$$\bar{\varphi}_{i,x}^{k,c} := \varphi_{i,x}^{k,c}(\mathbf{x}) + \frac{\partial \varphi_{i,x}^{k,c}(\mathbf{x})}{\partial z^k(\mathbf{x})} \cdot dz^k(\mathbf{x})\,, \tag{3.26}$$
$$\bar{\varphi}_{i,y}^{k,c} := \varphi_{i,y}^{k,c}(\mathbf{x}) + \frac{\partial \varphi_{i,y}^{k,c}(\mathbf{x})}{\partial z^k(\mathbf{x})} \cdot dz^k(\mathbf{x})\,. \tag{3.27}$$

Moreover, to improve the performance in low-textured regions, we follow [158] and [216] and normalize the linearized constraints. To this end, we introduce the following normalization factors

$$\theta_0^{k,c} := \left(\left(\frac{\partial \varphi_i^{k,c}(\mathbf{x})}{\partial z^k(\mathbf{x})} \right)^2 + \zeta^2 \right)^{-1}\,, \tag{3.28}$$

$$\theta_x^{k,c} := \left(\left(\frac{\partial \varphi_{i,x}^{k,c}(\mathbf{x})}{\partial z^k(\mathbf{x})} \right)^2 + \zeta^2 \right)^{-1}, \tag{3.29}$$

$$\theta_y^{k,c} := \left(\left(\frac{\partial \varphi_{i,y}^{k,c}(\mathbf{x})}{\partial z^k(\mathbf{x})} \right)^2 + \zeta^2 \right)^{-1}, \tag{3.30}$$

where $\zeta = 0.01$ is a small parameter to prevent division by zero. Combining linearization and normalization, we finally obtain the following differential formulation for the differential stereo data term

$$D_{\text{stereo}}^k\left(dz^k\right) = \frac{1}{n-1} \sum_{i=1}^{n-1} \int_{\Omega_0} \Psi_r \left(\sum_{c=1}^{3} \theta_0^{k,c} \left(\bar{\varphi}_{i,0}^{k,c} \right)^2 \right) + \gamma \cdot \Psi_r \left(\sum_{c=1}^{3} \theta_x^{k,c} \left(\bar{\varphi}_{i,x}^{k,c} \right)^2 + \theta_y^{k,c} \left(\bar{\varphi}_{i,y}^{k,c} \right)^2 \right) \mathrm{d}\mathbf{x} \,. \tag{3.31}$$

Differential SfS Data Term The derivation of the differential SfS data term turns out to be slightly more complicated. Due to its hyperbolic nature, we do not follow the standard procedure as in case of the stereo data term, but first replace the partial depth derivatives z_x and z_y that appear in the surface normal of the reflectance function $\mathbf{R}$ with a difference quotient based on an appropriate upwind scheme approximation [3], e.g., the one by Rouy and Tourin [146]. Employing the grid spacing h_x and h_y in x- and y-direction, respectively, the approximation for z_x is given as follows

$$\widetilde{z}_x = \max\left(\mathcal{D}^- z, -\mathcal{D}^+ z, 0\right), \tag{3.32}$$

$$\mathcal{D}^- z = \frac{z(x,y) - z(x-h_x,y)}{h_x}, \qquad \mathcal{D}^+ z = \frac{z(x+h_x,y) - z(x,y)}{h_x}, \tag{3.33}$$

where, for the simplicity of our presentation, we identify $z(\cdot,\cdot)$ with the corresponding grid values. Since the forward difference $\mathcal{D}^+ z$ enters Equation 3.32 with a negative sign, one has to restore the correct sign afterward via [35, 1]

$$z_x \approx \begin{cases} -\widetilde{z}_x & \text{if} \quad \widetilde{z}_x = -\mathcal{D}^+ z \,, \\ \widetilde{z}_x & \text{else} \,. \end{cases} \tag{3.34}$$

This approximation turns the dependency of $\mathbf{R}$ contained in the original SfS data term in Equation 3.10 on the local depth derivatives z_x, z_y into a dependency on those depth values from the neighborhood that are required to approximate these derivatives. In our case, this local neighborhood is given by the following five locations

$$z(\mathbf{x}+\mathbf{h}) \quad \text{with} \quad \mathbf{h} \in H = \{-\mathbf{h}_y, -\mathbf{h}_x, \mathbf{0}, +\mathbf{h}_x, +\mathbf{h}_y\} \tag{3.35}$$

where $\mathbf{h}_x = (h_x, 0)^\top$ and $\mathbf{h}_y = (0, h_y)^\top$ are the pixel offsets in x- and y-direction, respectively. Next, we use this approximation and follow the standard procedure. To this end, we introduce the incremental computation embedded in the coarse-to-fine scheme. Therefore, we linearize the

approximated reflectance model $\bar{\mathbf{R}}^k$ w.r.t. the increments dz^k, $\mathbf{dl}^k$, and $\mathbf{d}\boldsymbol{\rho}^k$. Please note that we adapt the employed grid spacing of the approximation h_x and h_y on each resolution level, which is denoted by the superscript k in the following. By introducing the following abbreviation

$$\phi^{k,c} := I_0^c - \bar{R}^{k,c}\,, \tag{3.36}$$

where the channel-wise entries $\bar{R}^{k,1}$, $\bar{R}^{k,2}$, and $\bar{R}^{k,3}$ of $\bar{\mathbf{R}}^k$ are computed using the known values z^k, $\mathbf{l}^k$, and $\boldsymbol{\rho}^k$ of level k, the linearized expression is given by

$$\begin{aligned}\bar{\phi}^{k,c}(\mathbf{x}) := \phi^{k,c}(\mathbf{x}) + \sum_{\mathbf{h}^k \in H^k} \frac{\partial \phi^{k,c}(\mathbf{x})}{\partial z^k(\mathbf{x}+\mathbf{h}^k)}\, dz^k(\mathbf{x}+\mathbf{h}^k) \\ + \left(\frac{\partial \phi^{k,c}(\mathbf{x})}{\partial \mathbf{l}^k(\mathbf{x})}\right)^\top \mathbf{dl}^k(\mathbf{x}) + \left(\frac{\partial \phi^{k,c}(\mathbf{x})}{\partial \boldsymbol{\rho}^k(\mathbf{x})}\right)^\top \mathbf{d}\boldsymbol{\rho}^k(\mathbf{x})\,.\end{aligned} \tag{3.37}$$

Take note that some of the depth derivatives, i.e., derivatives w.r.t. $z^k(\mathbf{x}+\mathbf{h}^k)$, are zero since the upwind scheme locally selects between a forward and a backward approximation and thus never uses all five depth values $z^k(\mathbf{x}+\mathbf{h}^k)$ from the neighborhood H^k to approximate the derivatives.

Finally, we can write the differential SfS data term as

$$D^k_{\text{sfs}}\left(dz^k, \mathbf{dl}^k, \mathbf{d}\boldsymbol{\rho}^k\right) = \int_\Omega \sum_{c=1}^{3} \left(\bar{\phi}^{k,c}\right)^2 \mathrm{d}\mathbf{x}\,. \tag{3.38}$$

Differential Regularization Terms Let us now discuss the differential formulations of the three smoothness terms. While the corresponding expressions for the anisotropic first-order regularizers for illumination and albedo are given by

$$R^k_{\text{illum}}\left(\mathbf{dl}^k\right) = \int_\Omega \sum_{l=1}^{2} \Psi_l\left(\left|\mathcal{J}(\mathbf{l}^k + \mathbf{dl}^k)\,\mathbf{r}_l\right|^2\right) \mathrm{d}\mathbf{x}\,, \tag{3.39}$$

$$R^k_{\text{albedo}}\left(\mathbf{d}\boldsymbol{\rho}^k\right) = \int_\Omega \sum_{l=1}^{2} g\left(\left|\mathcal{J}(\mathbf{ch}(\mathbf{I_0}))\,\mathbf{r}_l\right|^2\right) \cdot \Psi_l\left(\left|\mathcal{J}(\boldsymbol{\rho}^k + \mathbf{d}\boldsymbol{\rho}^k)\,\mathbf{r}_l\right|^2\right) \mathrm{d}\mathbf{x}\,, \tag{3.40}$$

the differential formulation of the anisotropic second-order regularizer for the depth reads

$$R^k_{\text{depth}}(dz^k) = \inf_{\mathbf{da}^k} \int_\Omega C^k(dz^k, \mathbf{da}^k) + \alpha_{\mathbf{a}} \cdot S^k(\mathbf{da}^k)\,\mathrm{d}\mathbf{x}\,, \tag{3.41}$$

with the following differential coupling and smoothness term

$$C^k(dz^k, \mathbf{da}^k) = \sum_{l=1}^{2} \Psi_l\left(\left(\mathbf{r}_l^\top\left(\nabla(z^k + dz^k) - (\mathbf{a}^k + \mathbf{da}^k)\right)\right)^2\right), \tag{3.42}$$

$$S^k(\mathbf{da}^k) = \sum_{l=1}^{2} \Psi_l\left(\sum_{m=1}^{2}\left(\mathbf{r}_m^\top \mathcal{J}(\mathbf{a}^k + \mathbf{da}^k)\,\mathbf{r}_l\right)^2\right). \tag{3.43}$$

Please note that also the auxiliary function $\mathbf{a}^k$ is computed incrementally, i.e. $\mathbf{a}^{k+1} := \mathbf{a}^k + \mathbf{da}^k$. In the coarse-to-fine scheme, the corresponding increments are updated jointly with the other increments, i.e., the increments for depth, illumination, and albedo.

INCREMENT REGULARIZATION Since the differential formulation of the energy uses an incremental linearization of the data terms, one has to ensure that the estimation is robust w.r.t. large erroneous increments that may arise in case the linearization is locally not valid. To this end, we penalize the length of the increments via

$$R_{\text{inc}}^k\left(dz^k, \mathbf{dl^k}, \mathbf{d}\boldsymbol{\rho}^{\boldsymbol{k}}\right) = \int_\Omega \alpha_{dz} \cdot \left|dz^k\right|^2 + \alpha_{\mathbf{dl}} \cdot \left|\mathbf{dl}^k\right|^2 + \alpha_{\mathbf{d}\boldsymbol{\rho}} \cdot \left|\mathbf{d}\boldsymbol{\rho}^k\right|^2 \mathrm{d}\mathbf{x}\,, \tag{3.44}$$

where α_{dz}, $\alpha_{\mathbf{dl}}$, and $\alpha_{\mathbf{d}\boldsymbol{\rho}}$ are weighting factors. Please note that the influence of the increment regularization vanishes as the incremental coarse-to-fine fixed point iteration converges because the regularizer only penalizes the increments and not the actual values. This statement particularly holds if one runs several iterations per resolution level as increments in later iterations tend to zero.

3.3.2 NUMERICAL SOLUTION

After deriving the differential formulation of the original energy, we can proceed as in our example described in Section 2.4.2 in order to minimize the differential energy at each resolution level. To this end, we first derive the necessary conditions for each minimizer in terms of the associated Euler-Lagrange equations and then provide details on how these equations can be solved numerically.

EULER-LAGRANGE EQUATIONS For the sake of clarity, we introduce the following abbreviations for the outer derivatives of the penalizer functions

$$\Psi'^k_{\text{stereo,bca}} := \Psi'_r\left(\sum_{c=1}^{3} \theta_0^{k,c}\left(\bar{\varphi}_{i,0}^{k,c}\right)^2\right), \tag{3.45}$$

$$\Psi'^k_{\text{stereo,gca}} := \Psi'_r\left(\sum_{c=1}^{3} \theta_x^{k,c}\left(\bar{\varphi}_{i,x}^{k,c}\right)^2 + \theta_y^{k,c}\left(\bar{\varphi}_{i,y}^{k,c}\right)^2\right), \tag{3.46}$$

$$\Psi'^k_{\text{illum}} := \Psi_l\left(\left|\mathcal{J}(\mathbf{l}^k + \mathbf{dl}^k)\,\mathbf{r}_l\right|^2\right), \tag{3.47}$$

$$\Psi'^k_{\text{albdeo}} := g\left(\left|\mathcal{J}(\mathbf{ch}(\mathbf{I_0}))\,\mathbf{r}_l\right|^2\right) \cdot \Psi'_l\left(\left|\mathcal{J}(\boldsymbol{\rho}^k + \mathbf{d}\boldsymbol{\rho}^k)\,\mathbf{r}_l\right|^2\right), \tag{3.48}$$

$$\Psi'^k_{\text{depth,c}} := \Psi'_l\left(\left(\mathbf{r}_l^\top(\nabla(z^k + dz^k) - (\mathbf{a}^k + \mathbf{da}^k))\right)^2\right), \tag{3.49}$$

$$\Psi'^k_{\text{depth,s}} := \Psi'_l\left(\sum_{m=1}^{2}\left(\mathbf{r}_m^\top \mathcal{J}(\mathbf{a}^k + \mathbf{da}^k)\mathbf{r}_l\right)^2\right), \tag{3.50}$$

as well as the following four diffusion tensors resulting from the anisotropic regularizers

$$\mathbf{T}_z^k := \sum_{l=1}^{2} \Psi'^k_{\text{depth,c}} \cdot \mathbf{r}_l \mathbf{r}_l^\top \,, \tag{3.51}$$

$$\mathbf{T}_{\mathbf{a}}^k := \sum_{l=1}^{2} \Psi'^k_{\text{depth,s}} \cdot \mathbf{r}_l \mathbf{r}_l^\top \,, \tag{3.52}$$

$$\mathbf{T}_{\mathbf{l}}^k := \sum_{l=1}^{2} \Psi'^k_{\text{illum}} \cdot \mathbf{r}_l \mathbf{r}_l^\top \,, \tag{3.53}$$

$$\mathbf{T}_{\boldsymbol{\rho}}^k := \sum_{l=1}^{2} \Psi'^k_{\text{albdeo}} \cdot \mathbf{r}_l \mathbf{r}_l^\top \,. \tag{3.54}$$

Using these abbreviations allows us to write the Euler-Lagrange equations associated with the differential energy given in Equation 3.22 in a more compact form. In this case, the Euler-Lagrange equations constitute a coupled system of nine non-linear partial differential equations that read

$$\begin{aligned} 0 = &\sum_{c=1}^{3} \Bigg(\tfrac{1}{n-1} \sum_{i=1}^{n-1} \Bigg(\Psi'^k_{\text{stereo,bca}} \cdot \Big(\theta_0^{k,c}\, \bar{\varphi}_{i,0}^{k,c}\, \tfrac{\partial \varphi_i^{k,c}(\mathbf{x})}{\partial_{z^k}(\mathbf{x})} \Big) + \gamma \cdot \Psi'^k_{\text{stereo,gca}} \cdot \Big(\theta_x^{k,c}\, \bar{\varphi}_{i,x}^{k,c}\, \tfrac{\partial \varphi_{i,x}^{k,c}(\mathbf{x})}{\partial z^k(\mathbf{x})} \\ &+ \theta_y^{k,c}\, \bar{\varphi}_{i,y}^{k,c}\, \tfrac{\partial \varphi_{i,y}^{k,c}(\mathbf{x})}{\partial z^k(\mathbf{x})} \Big) \Bigg) + \sum_{\mathbf{m}^k \in H^k} \Big(\bar{\phi}^{k,c}(\mathbf{x} + \mathbf{m}^k)\, \tfrac{\partial \phi^{k,c}(\mathbf{x}+\mathbf{m}^k)}{\partial z^k(\mathbf{x})} \Big) \Bigg) \\ &- \alpha_z \cdot \operatorname{div} \Big(\mathbf{T}_z^k \Big(\nabla(z^k + dz^k) - (\mathbf{a}^k + \mathbf{da}^k) \Big) \Big) + \alpha_{dz} \cdot dz^k \,, \end{aligned} \tag{3.55}$$

$$\mathbf{0} = \sum_{c=1}^{3} \Big(\bar{\phi}^{k,c}(\mathbf{x}) \cdot \tfrac{\partial \phi^{k,c}(\mathbf{x})}{\partial \mathbf{l}^k(\mathbf{x})} \Big) - \alpha_{\mathbf{l}} \cdot \mathbf{div} \Big(\mathcal{J}(\mathbf{l}^k + \mathbf{dl}^k)\, \mathbf{T}_{\mathbf{l}}^k \Big) + \alpha_{\mathbf{dl}} \cdot \mathbf{dl}^k \,, \tag{3.56}$$

$$\mathbf{0} = \sum_{c=1}^{3} \Big(\bar{\phi}^{k,c}(\mathbf{x}) \cdot \tfrac{\partial \phi^{k,c}(\mathbf{x})}{\partial \boldsymbol{\rho}^k(\mathbf{x})} \Big) - \alpha_{\boldsymbol{\rho}} \cdot \mathbf{div} \Big(\mathcal{J}(\boldsymbol{\rho}^k + \mathbf{d}\boldsymbol{\rho}^k)\, \mathbf{T}_{\boldsymbol{\rho}}^k \Big) + \alpha_{\mathbf{d}\boldsymbol{\rho}} \cdot \mathbf{d}\boldsymbol{\rho}^k \,, \tag{3.57}$$

$$\mathbf{0} = \mathbf{T}_z^k \cdot \Big(\mathbf{a}^k + \mathbf{da}^k - \nabla\Big(z^k + dz^k \Big) \Big) - \alpha_{\mathbf{a}} \cdot \mathbf{div} \Big(\mathcal{J}(\mathbf{a}^k + \mathbf{da}^k)\, \mathbf{T}_{\mathbf{a}}^k \Big) \,, \tag{3.58}$$

where the $\mathbf{div}$ operator applies the standard divergence operator div to the rows of a matrix-valued function, e.g.,

$$\mathbf{div} \Big(\mathcal{J}(\mathbf{l}^k + \mathbf{dl}^k)\, \mathbf{T}_{\mathbf{l}}^k \Big) = \begin{pmatrix} \operatorname{div}\big(\mathbf{T}_{\mathbf{l}}^k\, \nabla(l_1^k + dl_1^k) \big) \\ \operatorname{div}\big(\mathbf{T}_{\mathbf{l}}^k\, \nabla(l_2^k + dl_2^k) \big) \\ \operatorname{div}\big(\mathbf{T}_{\mathbf{l}}^k\, \nabla(l_3^k + dl_3^k) \big) \end{pmatrix} . \tag{3.59}$$

In contrast to the general Euler-Lagrange equations with second-order derivatives given in the foundation chapter, i.e., in Subsection 2.4.1, the previous Euler-Lagrange equations contain non-local contributions, due to the approximation of the hyperbolic warping scheme. A more detailed derivation of the Euler-Lagrange equations can be found in the appendix of [4].

Discretization and Numerical Solution In order to solve the Euler-Lagrange equations (3.55)–(3.58) at each resolution level, we discretize them on a rectangular grid. The corresponding grid spacing h_x and h_y is derived from the given camera calibration matrix together with the focal length. Please note that the pixel size increases on coarser resolutions. Hence, as mentioned before, the camera calibration matrices K_i^k and the projection matrices P_i^k have to be adapted accordingly at each resolution level of the coarse-to-fine minimization.

Regarding the discretization of the stereo data term, we compute the derivatives of $\varphi_i^{k,c}$ in the linearized expression analytically, see Section A.1 for additional information. We discretize the occurring depth derivatives in this process employing standard finite differences. Furthermore, we compute expressions of type $I_i^c(\mathbf{x}_i^k)$ via warping, see Section A.1 for details on how to adapt the warping to the stereo scenario. We refer to this part of the optimization as *geometric warping* [36].

In the case of the SfS data term, we compute the derivatives of $\phi^{k,c}$ in the linearized data term numerically. To this end, we vary the current estimates z^k, $\mathbf{l}^k$, and $\boldsymbol{\rho}^k$ by $\pm 10^{-12}$ and re-evaluate the expressions, which allows computing the derivatives w.r.t. the different unknowns with a standard central difference scheme. Please recall in this context the values of $\phi^{k,c}$ required for computing these derivatives are evaluated based on an upwind approximation of the depth derivatives in the reflectance function $\bar{\mathbf{R}}^k$. Hence, at each level, before estimating the desired increments, we do not only have to evaluate the values $\phi^{k,c}$ based on the current depth z^k, but we also have to decide whether forward or backward approximations are locally used within the upwind scheme. Consequently, we term this part of the optimization *hyperbolic warping* [3].

Furthermore, we have to discretize the divergence expressions resulting from the regularization terms for depth, illumination, and albedo. In this context, we make use of the advanced discretization scheme proposed by Weickert et al. [186].

Finally, we have to solve the resulting non-linear system of equations. To this end, we proceed as in our example shown in Subsection 2.4.2 and employ a second fixed-point iteration, where we keep all the remaining non-linear expressions, i.e., the outer derivatives (3.45)–(3.50) of the sub-quadratic penalizer functions fixed. The resulting linear systems of equations are then solved using the SOR method [205].

Implementation Details Let us finally comment on four important implementation details: the number of outer fixed point steps per resolution level, the initialization of the depth, illumination and albedo, the resolution dependent weighting between the data terms for SfS and stereo, and the level depending adjustment of the amount of regularization.

(i) So far, we have assumed that we perform a single linearization per resolution level. Since we regularize the length of the increments, however, a single linearization per resolution level is not sufficient. Hence, in our final algorithm, we perform several fixed point iterations per resolution level which significantly improves the reconstruction quality. (ii) Regarding the initialization, we use the following rather intuitive strategy. At the coarsest level, we initialize the depth z with a fronto-parallel plane, such that $\nabla z = \mathbf{a} = \mathbf{0}$, the illumination vector $\mathbf{l}$ with zero (not to prefer any particular direction), and the albedo $\boldsymbol{\rho}$ with the downsampled input image. (iii) Furthermore, to account for the fact that the zero initialization of the illumination vector does not allow the SfS data term to provide any useful information at coarser levels, we introduce a sigmoid weighting

function that increases the SfS weight ν towards finer levels. The corresponding weight on the resolution level k is given by

$$\nu^k := s_\nu^k \cdot \nu\,, \quad \text{with} \quad s_\nu^k := \frac{1}{1 + e^{\frac{-(k/k_{\max})+b}{a}}}\,, \tag{3.60}$$

where $k_{\max}$ denotes the total number of levels and a and b are parameters that allow for adjusting the slope and the shift of the sigmoid function, respectively. Throughout our experiments we set $a = 0.1$ and $b = 0.5$ fixed. Please note that we apply the same scaling to the albedo and illumination regularization weights to ensure that the relative weighting between the different SfS-related terms is not affected. (iv) Finally, we employ a level depended scaling of α_z, $\alpha_{\mathbf{l}}$, and $\alpha_{\boldsymbol{\rho}}$ using the following scale factor

$$\alpha_z^k := s_\alpha^k \cdot \alpha_z\,, \quad \alpha_{\mathbf{l}}^k := s_\alpha^k \cdot \alpha_{\mathbf{l}}\,, \quad \alpha_{\boldsymbol{\rho}}^k := s_\alpha^k \cdot \alpha_{\boldsymbol{\rho}} \tag{3.61}$$

with

$$s_\alpha^k := \sqrt{h_x^k \cdot h_y^k}\,, \tag{3.62}$$

where h_x^k and h_y^k denote the grid spacing of the current resolution level k in x- and y-direction, respectively. This strategy reduces the effect of the regularization at finer levels compared to coarser levels. As a consequence, it allows preserving significantly more details in the reconstruction while still avoiding to get trapped in local minima at coarser levels.

3.4 Evaluations

In this section, we evaluate the introduced model that exploits parallax and shading cues simultaneously within a joint approach. We analyze the model quantitatively and compare it to a variant that only uses parallax cues as well as other stereo methods from the literature. Furthermore, we oppose it qualitatively to other shading cue based approaches.

Evaluation Setup In all experiments, the following fixed set of solver-related parameters is used: a downsampling factor of $\eta = 0.8$, 20 iterations per resolution level, 2 non-linear fixed-point iterations, and 20 SOR iterations with an over-relaxation parameter of $\omega = 1.8$. Using this parameter set the runtime of the non-optimized C++ implementation is in the order of 1h 40m when applied to input images of size 1536×1024 and run on a single core with 3.40 GHz (Intel Core i7-2600 CPU). The remaining parameters are specified in the appendix, see Section B.1.

For the experiments, we consider synthetic and real-world data. The utilized *synthetic data* was created in Blender [30] using the Blunderbuss Pete model[1] with artificial procedural Voronoi texturing. It consists of three views, and two distant light sources illuminate the scene. The utilized *real-world data* consists of indoor and outdoor scenes. In total it consists of four data sets: the Angel data set of Wu et al. [194] (5 views), the Socrates data set of [218] (7 views) and the Fountain (2 views) and Herz-Jesu (2 views) data set of [161]. All data sets comprise fine surface details and subtle geometry which pose a challenging task for 3D surface reconstruction. The reference view, as well as an example match view of all the considered data sets, can be found in Figure 3.4.

[1] by Ben Dansie (`www.thingiverse.com/thing:144775`)

Figure 3.4: Reference view (top row) and one match view (bottom row) of the considered data sets. *From left to right*: Blunderbuss Pete, Angel, Fountain, Herz-Jesu, and Socrates.

3.4.1 Synthetic Data

Constraint Normalization In the first experiment, we analyze the influence of the constraint normalization, employed in the differential stereo data term. To this end, we consider a pure stereo variant of our method which we obtain by omitting the SfS data term as well as the illumination and albedo regularization terms. Figure 3.5 depicts the results obtained for the Blunderbuss Pete data set with and without constraint normalization. They reveal that without normalization artifacts arise at image edges – even if one increases the amount of regularization; see the middle image of Figure 3.5. This finding is in accordance with [216], who proposed such a normalization in the context of motion estimation.

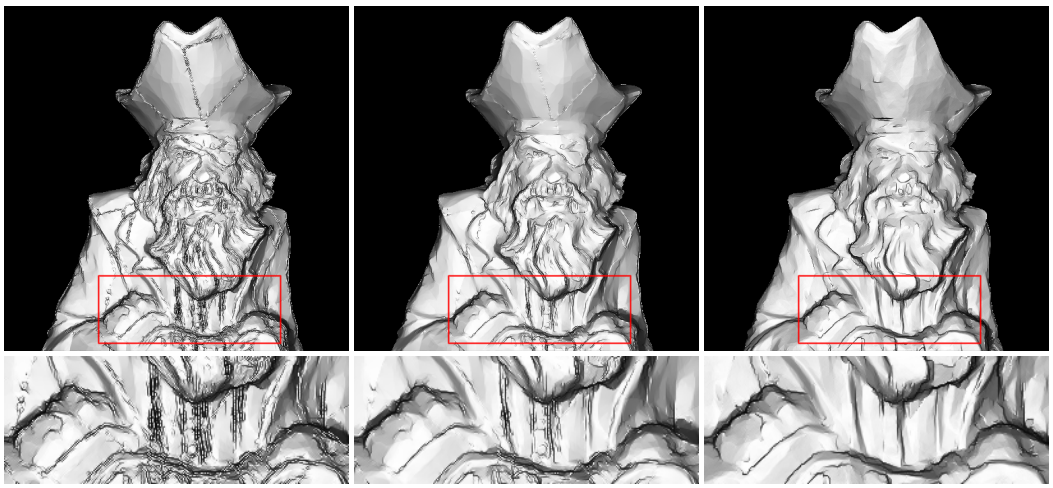


Figure 3.5: Synthetic Blunderbuss Pete data set (3 views). Influence of the constraint normalization in the data term of a pure stereo method. *From left to right*: Without constraint normalization, without constraint normalization but increased smoothness, and with constraint normalization.

Stereo vs. Combined Approach In the second experiment, we investigate the benefit of exploiting depth and parallax cues simultaneously. Therefore, we compare the reconstruction quality of our combined approach with the pure stereo variant. Figure 3.6 shows the reference image, the ground truth as well as the results for the pure stereo method and the combined approach. Moreover, it also depicts the estimated albedo and the computed illumination direction of the combined approach. As one can see, the combined approach reconstructs fine surface details such as the eye and the beard much better than the pure stereo method that yields a somewhat coarser result with sporadic artifacts. Furthermore, the estimated albedo and the computed illumination direction in Figure 3.6 look quite reasonable. Thus it is not surprising that the clear visual improvement in small surface details is also confirmed quantitatively by a slight decrease of the root mean square (RMS) error of the surface from $19.52 \cdot 10^{-5}$ to $19.14 \cdot 10^{-5}$. In this context, one has to keep in mind that the improvement lies mainly in the reconstruction of small surface details. Regarding the average angular error (AAE) of the surface normals, the improved becomes even more explicit. Here, the error decreases from $18.57°$ to $17.52°$.

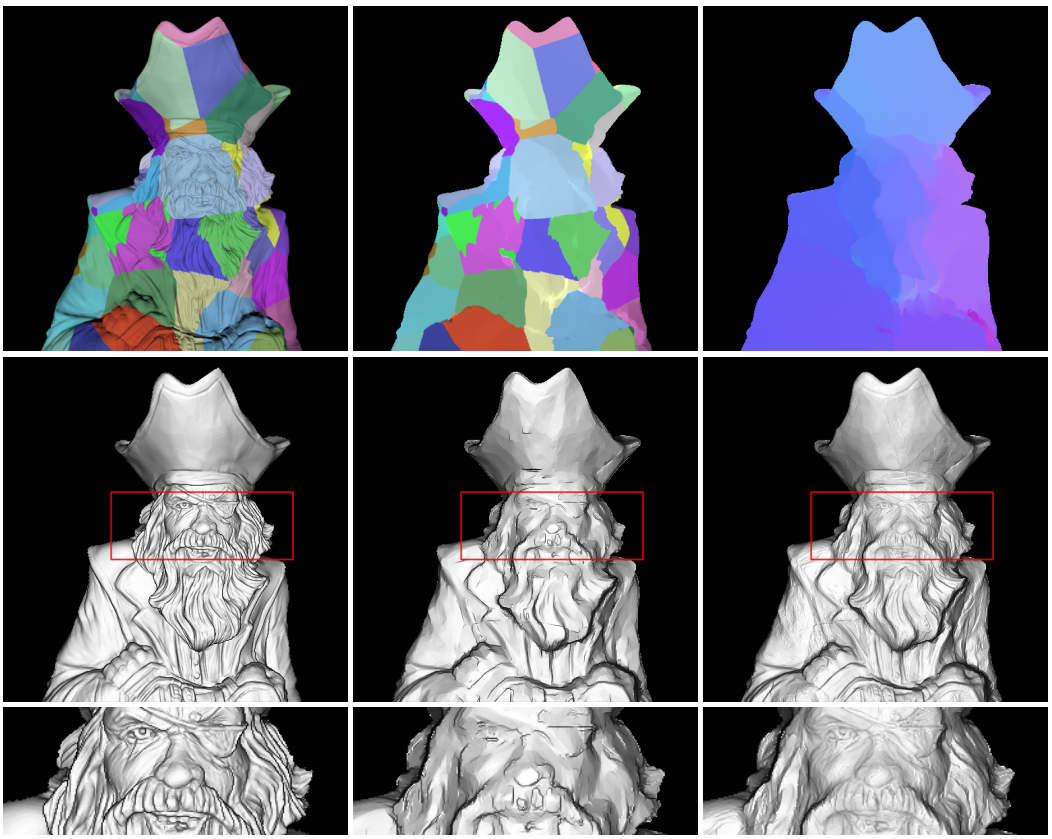


Figure 3.6: Synthetic Blunderbuss Pete data set. Three-view results. *First row, from left to right*: Reference input image, computed albedo, computed illumination direction. *Second and third row, from left to right*: Shaded images showing the ground truth, pure stereo and our combined approach.

3.4.2 Real-World Data

Comparison to Sequential Methods For the first real-world experiment, we use the Angel data set. Once again, we compute results for the pure stereo variant and the combined approach. Moreover, we added the results of the approach of Wu et al. [194] for comparison – a method that refines a pre-computed multi-view stereo mesh using shading information. Once more, the corresponding reconstructions in Figure 3.7 show that the pure stereo variant is not able to capture all fine-scale details such as the strands of hair, the disc area of the sunflower head or the toes. The method of Wu et al. does better. However, the overall reconstruction is too smooth. In particular, coarse structures such as the sunflower petals pointing towards the camera or the ringlet are over-smoothed. In contrast, our combined approach can recover both coarse-scale and fine-scale details accurately. This observation becomes apparent when comparing our results to the reference image.

Apart from the Angel data set, we also consider the Socrates data set. Doing so allows providing a visual comparison to the shading-based refinement method of Zollhöfer et al. [218] – an approach that operates on implicit surfaces in terms of volumetric signed distance functions. Figure 3.8 shows

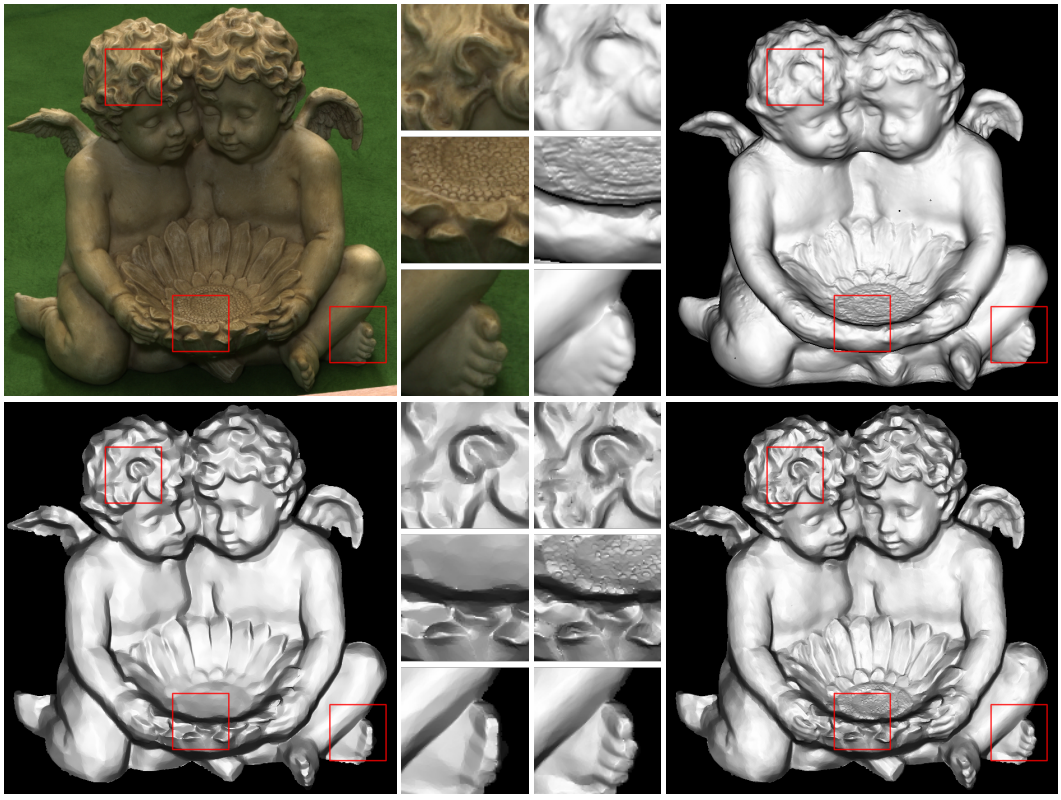


Figure 3.7: Real-world Angel data set [194]. Five-view results. *Top left*: Reference image. *Top right*: Sequential method of Wu et al. [194]. *Bottom left*: Our pure stereo approach. *Bottom right*: Our combined approach.

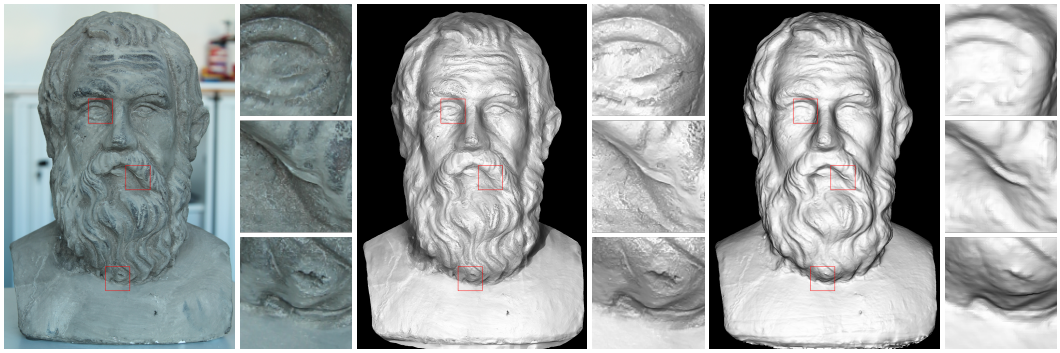


Figure 3.8: Real-world Socrates data set [218]. Seven-view results. *From left to right*: Reference image, our combined approach, sequential method of Zollhöfer et al. [218].

the corresponding results. As one can see both methods provide visually appealing reconstructions. While the result from Zollhöfer et al. is slightly smoother, our combined approach recovers more details; see for instance the eyes, the beard or small damages of the sculpture.

Comparison to Stereo Methods For our second real-world experiment, we used the Fountain and the Herz-Jesu data sets for which an approximate ground truth captured with a time of flight laser system is available [161]. Since the recovery of fine details strongly depends on the sharpness of the input data, we downsampled the slightly blurred images to half the resolution before reconstructing the scenes from only two views. This time, apart from the results of our combined method and its stereo variant, we also provide results for two recently proposed stereo approaches which are able to handle arbitrary camera settings and which provide source code publicly: On the one hand, we use the variational method of Graber et al. [69] that uses a minimal-surface regularization. For the given data set this method has shown significant improvements compared to standard TV regularization. On the other hand, we consider the basic approach of Galliani et al. [62] which is a multi-view variant of PatchMatch Stereo [31]. While Galliani et al. also proposed an additional 3D integration step in terms of fusing multiple reconstructions from different views, we had to omit this step in our experiment, since we are not interested in a closed reconstruction but in evaluating the quality of the depth map from the reference camera.

Figure 3.9 depicts qualitative results for the Fountain. While the multi-view PatchMatch method of Galliani et al. recovers significant jumps very accurately, the corresponding reconstruction lacks fine details and contains significant outliers in occluded regions. The latter observation is a direct consequence of the lacking regularization of the PatchMatch algorithm. In contrast, the approach of Graber et al. yields a more detailed reconstruction that is, however, very noisy. This noise, in turn, is a consequence of the minimal-surface regularization that tends to round-off objects when suppressing local fluctuations and thus only preserves surface details if the amount of regularization is chosen sufficiently low.

In comparison, the reconstruction of our pure stereo method is already quite accurate. While flat surfaces are almost noise free, details of the fountain and the wall are more pronounced. These facts indeed show the benefits of the edge-preserving anisotropic second-order regularization. The visually most appealing reconstruction, however, for this data set is obtained by our combined

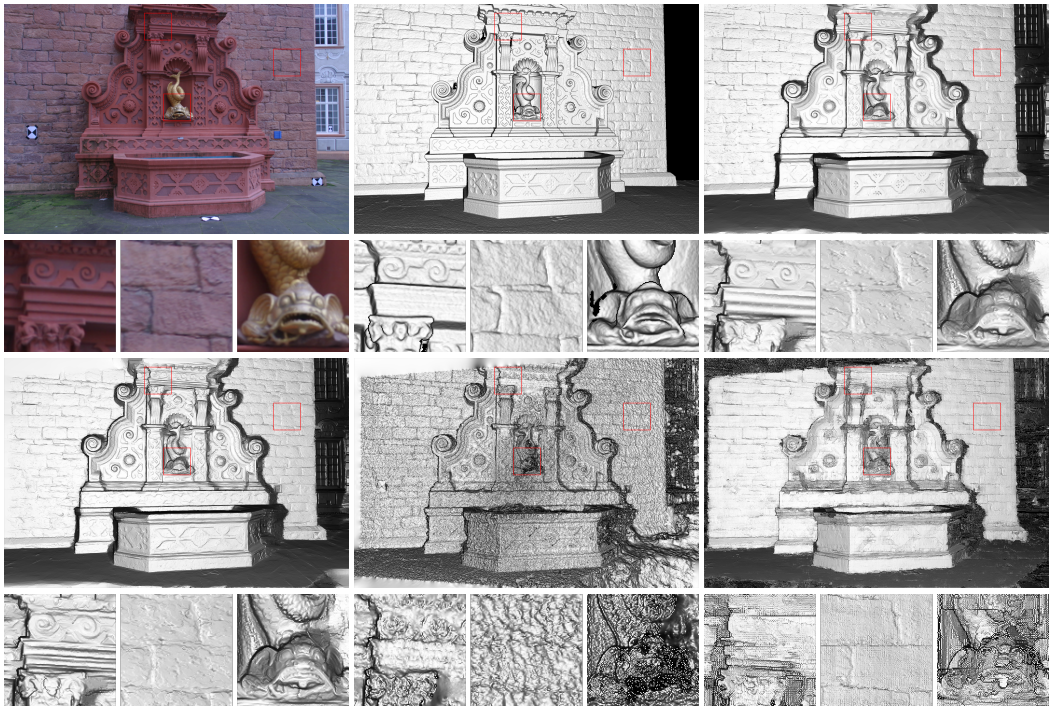


Figure 3.9: Real-world Fountain data set [161]. Two-view results. *Top left*: Reference image. *Top center*: Ground truth. *Top right*: Our combined approach. *Bottom left*: Our pure stereo approach. *Bottom center*: Graber et al. [69]. *Bottom right*: Galliani et al. [62].

Table 3.1: Comparison to stereo methods in terms of the root mean square (RMS) error of the surface and the average angular error (AAE) of the surface normals for the Fountain and Herz-Jesu data set.

	Fountain				Herz-Jesu			
	RMS		AAE		RMS		AAE	
method	all	non-occ.	all	non-occ.	all	non-occ.	all	non-occ.
Graber et al. [69]	0.0688	0.0367	40.76°	39.41°	0.2217	0.0535	43.82°	42.45°
Graber et al. [69]	0.0264[1]	–	–	–	–	–	–	–
Galliani et al. [62]	0.6124	0.0157	27.65°	21.98°	3.2813	0.9632	40.30°	34.79°
Ours (stereo)	0.0168	0.0023	18.88°	16.34°	0.0706	0.0328	21.82°	19.96°
Ours (stereo + SfS)	**0.0134**	**0.0022**	**16.91°**	**14.92°**	**0.0666**	**0.0321**	**21.23°**	**19.28°**

[1] While the publicly available Python code does not achieve such low errors – even with optimized parameters – they have been reported for the non-publicly available CUDA code for the full resolution images; see [69].

approach. It recovers even fine-scale details such as the mouth of the fish and the ornaments of the fountain. This result, in turn, demonstrates the usefulness of additional shading information. One makes similar observations in the case of the results for the Herz-Jesu data set provided in Figure 3.10. Also there, the combined method shows the most appealing results visually. Table 3.1 confirms our findings by a quantitative comparison of the results. It shows that the RMS and AAE errors of our methods are significantly lower than those of the other two approaches both for the Fountain as well as for the Herz-Jesu data set.

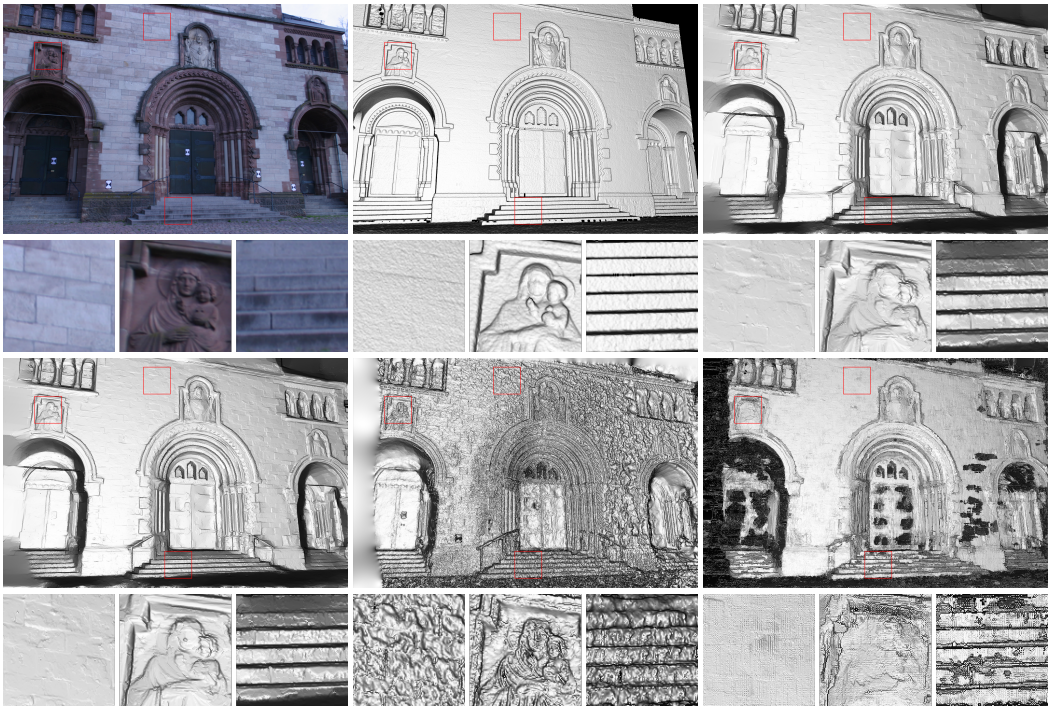


Figure 3.10: Real-world Herz-Jesu data set [161]. Two-view results. *Top left*: Reference image. *Top center*: Ground truth. *Top right*: Our combined approach. *Bottom left*: Our pure stereo approach. *Bottom center*: Graber et al. [69]. *Bottom right*: Galliani et al. [62].

Table 3.2: Comparison to the partially isotropic variant of our conference paper [5] in terms of the root mean square (RMS) error of the surface and the average angular error (AAE) of the surface normals for the Fountain and Herz-Jesu data set.

	Fountain				Herz-Jesu			
	RMS		AAE		RMS		AAE	
method	all	non-occ.	all	non-occ.	all	non-occ.	all	non-occ.
partially isotropic model	**0.0134**	**0.0022**	17.31°	15.28°	0.0695	0.0325	**20.98°**	**19.02°**
our anisotropic model	**0.0134**	**0.0022**	**16.91°**	**14.92°**	**0.0666**	**0.0321**	21.23°	19.28°

COMPARISON TO AN ISOTROPIC MODEL In our final experiment we compare the results of our anisotropic model, which employs anisotropic regularization for all the unknowns (depth, illumination, and albedo), with the partially isotropic model as presented in our conference paper in [5], which employs an anisotropic regularization only in the context of the depth but isotropic regularization in case of the illumination and the albedo. For this purpose, we show the albedo as well as the direction of the illumination vector obtained for the Fountain and the Herz-Jesu data set in Figure 3.11 and Figure 3.12, respectively. As one can observe, our anisotropic model allows capturing sharper, better-aligned edges, which is especially beneficial at object boundaries. A notable decrease in terms of the RMS error for the Herz-Jesu data set given in Table 3.2 confirms this visual improvement. However, it is also accompanied by a slight increase in terms of the AAE

which might be related to the fact, that the results have been optimized for the RMS error. In case of the already accurate estimate for the Fountain, the RMS error remained the same, but here the AAE improved notably.

Figure 3.11: Real-world Fountain data set [161]. *Top to bottom*: Computed illumination direction and computed albedo. *Left to right*: Partially isotropic model [5] and our anisotropic model.

Figure 3.12: Real-world Herz-Jesu data set [161]. *Top to bottom*: Computed illumination direction and computed albedo. *Left to right*: Partially isotropic model [5] and our anisotropic model.

3.5 Limitations

Although the proposed algorithm provided excellent results in our evaluation, it also has some limitations that one could address in future work. On the one hand, specular reflections and other effects, e.g., roughness and transparency, violate the Lambertian assumption. One could tackle this problem either by using a non-Lambertian reflectance model that can model more realistic reflections, e.g., [95, 117, 178], or by using dedicated error variables that can capture the violations implicitly, e.g., [113]. On the other hand, the model is based on variational optimization and thus mainly suitable for small baselines. While this does not pose a problem when the user has full control over the image acquisition pipeline, integrating additional information such as feature matches could help to overcome this limitation. Finally, from a robustness viewpoint, also a final integration of different reconstructions from multiple viewpoints seems desirable [62, 152, 211]. This integration would allow to rule out inconsistencies, to address occlusions more explicitly and to average out possible noise in the reconstruction.

3.6 Conclusions

In this chapter, we have proposed a novel view-centered variational method that combines stereo and shape from shading. In this context, our contribution was fivefold: (i) We showed how shading and disparity information could be integrated explicitly into a joint minimization framework for estimating the depth. In contrast to most existing approaches, we thereby refrained from using any form of stereo-based pre-estimation. (ii) We made use of an adaptive anisotropic second-order smoothness term. This term further encouraged the detail-preserving reconstruction of non-fronto-parallel surfaces. (iii) We extended this model in such a way that it additionally allows to estimate albedo and illumination. This extension made our approach applicable to more general scenarios including Lambertian objects with non-uniform albedo and scenes with unknown illumination. (iv) In this context, we also made use of anisotropic regularization. This choice, in turn, allowed the estimation of detailed albedo and illumination maps. (v) Finally, we derived a coarse-to-fine minimization framework based on a linearization of all data terms. This linearization not only enabled the application of standard optimization techniques such as nested fixed point iterations, but it also allowed the joint estimation of all unknowns. Experiments for synthetic and real-world images demonstrate that our combined approach allows for accurate and detailed reconstructions. Moreover, they show that shading information is indeed useful to improve upon pure stereo methods, in particular when it comes to the reconstruction of small-scale details. Finally, they also indicate that the strategy of jointly estimating all unknowns may be indeed worthwhile. Compared to sequential refinement approaches, it became possible to obtain reconstructions that were slightly more detailed.

4 Variational Motion Estimation

Starting with the seminal work of Horn and Schunck [81], variational methods dominated the field of motion estimation for several decades, since they not only allow for transparent modeling but also offer dense and accurate results. In order to compute a flow field, such methods minimize a so-called cost or energy functional, which constitutes a measure of correctness w.r.t. certain assumptions. In the original formulation, this energy functional is composed of two terms: a data term that imposes temporal constancy constraints on image features and a regularization term that enforces spatial regularity on the solution. While the data term enables to trace corresponding points in subsequent frames, the regularization term allows coping with ill-posed situations. In particular, the regularization term usually models some sort of smoothness assumption that enables the so-called filling-in effect: at locations where no reliable local flow estimate is possible the regularizer fills in information from the neighborhood. This effect can be analyzed by examining the underlying diffusion process.

In this chapter, we focus on the regularization component for variational optical flow estimation. In the first part of this chapter, i.e., Section 4.1, we compare different strategies on how to model isotropic and anisotropic second-order regularizers. In this context, we not only consider existing regularizers but pursue a systematic course of action and include new techniques that have not been considered so far. Furthermore, we analyze the underlying diffusion processes of the different regularizers to gain a better understanding of the exhibited anisotropy. In the second part of this chapter, i.e., Section 4.2, we improve upon fixed-order regularization and propose a new order-adaptive regularizer that allows to combine benefits of first and second-order regularization. To achieve this we resort to adequate regularizers that have been identified in the first part of this chapter and develop a sophisticated approach to link them. Main parts of this chapter are based on the work published in [9, 10].

4.1 Comparison of Second-Order Regularizers

As already mentioned regularization plays a key role within variational motion estimation since it allows to cope with the ill-posed nature of the problem. While many variational motion estimation methods rely on first-order regularization strategies [36, 124, 162, 193, 217] which assume mainly fronto-parallel motion, approaches based on second-order regularization have gained more and more attention [33, 54, 80, 137, 167]. In particular in scenes with a vast amount of ego-motion, such second-order regularizers allow to estimate the resulting piecewise affine flow fields which cannot be captured adequately by first-order regularizers.

4.1.1 Related Work

In the following, we give an overview of existing strategies to model second-order smoothness assumptions. Thereby, the focus is on local regularization strategies, i.e., techniques that do not use larger neighborhoods; cf. [137]. Furthermore, also techniques proposed in a non optical flow context are discussed. In general, one can divide the considered modeling strategies into three different classes: direct approaches, combined approaches, and indirect approaches.

Direct Approaches Probably the most intuitive way to model second-order smoothness assumptions is to penalize second-order derivatives of the unknown functions directly. Therefore, we refer to the corresponding methods as *direct approaches*. Such approaches include for example the Hessian [54,107,115,149,177], the Laplacian [47,115] and operators based on decorrelated second-order derivatives [167]. However, while being able to capture affine flow fields such methods also have a decisive drawback. They do not allow to model discontinuities in the first-order derivatives and hence do not preserve jumps equally well as first-order approaches.

Combined Approaches To tackle the problem of not being able to model jumps, one can combine direct second-order regularizers with appropriate first-order counterparts. This combination can be achieved in two ways. On the one hand, one can apply both regularizers at the same time. Thereby, one can realize switching with a spatially adaptive weight [109]. On the other hand, one can additively split the unknowns into two or more layers and apply a separate regularization to each of the layers in terms of an infimal convolution [45]. This splitting, however, requires to cope with additional unknowns and a sophisticated weighting strategy.

Indirect Approaches Indirect approaches constitute a third class, which includes coupling models. These coupling models typically realize second-order smoothness assumptions by introducing auxiliary functions which approximate first-order derivatives, and by using regularizers that enforce smoothness assumptions on these auxiliary functions. This strategy, in turn, allows to model discontinuities in both first and second-order derivatives. Such methods include, e.g., the total generalized variation (TGV) [33, 34] and its variants [58, 75, 80, 136, 138]. To the same group, one may also count over-parametrized approaches [128], which approximate a second-order regularization by introducing an affine parametrization of the unknowns and using a first-order regularization of the coefficients. However, as shown in [167] such a parametrization treats jumps at different locations differently and hence may not lead to the desired second-order regularization, since the inferred solutions rarely are piecewise constant in practice.

A second important concept in the context of modeling smoothness terms apart from considering higher order derivatives is the use of directional information. Such anisotropic strategies have proven to be beneficial not only in the context of first-order regularizers [124, 162, 193, 216] but also w.r.t. second-order regularizers [58, 75, 109, 138]. Lenzen et al. [109] embedded such concepts in a combined approach, where image information is used to steer the directions. Regarding indirect approaches, Ranftl et al. [138], Ranftl [136] and Ferstl et al. [58] introduced similar concepts into the coupling term, which connects the auxiliary functions and the first-order derivatives. Recently, Hafner et al. [75] extended this work by applying the anisotropy not only in the coupling term but also in the smoothness term, which enforces directional smoothness on the auxiliary functions.

Please note that from all the second-order approaches mentioned above only the works [33, 54, 80, 128, 136, 167] have been proposed in the context of optical flow estimation, and only one of

them [136] makes use of anisotropic strategies. However, the strategy in [136] limits the anisotropy to certain components and hence does not exploit the full potential of directional adaptation. Moreover, as observed by Lellmann et al. [108], introducing anisotropic concepts in higher order smoothness terms allows choosing different penalization strategies related to varying degrees of anisotropy – in contrast to the first-order case. Hence, not only the question persists which of the three classes mentioned above performs the best in the context of optical flow estimation, but also which degree of anisotropy is most suitable when modeling second-order smoothness terms for this task.

4.1.2 Contributions

In the first part of this chapter, i.e., Section 4.1, we address both these questions. On the one hand, we investigate and compare representative approaches of all three above mentioned classes and demonstrate the benefits of introducing anisotropic concepts in each of these classes. In this context, we also systematically analyze the diffusion processes induced by the different regularization strategies by using a convenient notation. On the other hand, we propose a novel anisotropic second-order regularization strategy. This new strategy exceeds existing optical flow regularizers regarding the degree of anisotropy. To evaluate the different strategies, we consider two popular benchmarks: the KITTI 2012 [65] and KITTI 2015 benchmark [119]. Both of these benchmarks contain a vast amount of ego-motion and thus are relevant for our analysis. The conducted experiments not only show that an indirect approach in terms of a coupling model is favorable but also that the concept of integrating direction information consistently improves the results.

4.1.3 Baseline Model

To compare different regularization techniques in the context of variational optical flow estimation, we need to embed them into a variational model. Hence, we next introduce an optical flow approach that will serve as a baseline model. When choosing such model, one has to keep in mind that typical real-world image sequences, e.g., the KITTI benchmarks, contain not only a large amount of non-fronto parallel motion but also illumination changes. Demetz et al. [54] proposed a method that tackles the challenge of handling such illumination changes. This approach explicitly models illumination changes in terms of a set of coefficient fields. Therefore, we employ their model as a baseline for our prototypes in this section.

Given two consecutive image frames $I_1, I_2 : \Omega \to \mathbb{R}$ of an image sequence, the method seeks to compute both the flow field $\mathbf{w} = (u, v)^\top : \Omega \to \mathbb{R}^2$ and the set of coefficient fields $\mathbf{c} = (c_1, \dots, c_n)^\top : \Omega \to \mathbb{R}^n$ as the minimizer of the following energy functional:

$$E(\mathbf{w}, \mathbf{c}) = \int_\Omega D(\mathbf{w}, \mathbf{c}) + \alpha \cdot R_{\text{flow}}(\mathbf{w}) + \beta \cdot R_{\text{illum}}(\mathbf{c})\, d\mathbf{x}\,, \tag{4.1}$$

where $\mathbf{x} = (x, y)^\top \in \Omega$ denotes the location within the rectangular image domain Ω. The energy functional consists of a data term D and two regularization terms R_{flow} and R_{illum} for the flow field and for the coefficient fields, respectively. Moreover it makes use of two weighting parameters α and β to allow balancing the relative impact of all three terms.

DATA TERM Let us now take a closer look at the data term. Classical data terms for variational motion estimation seek to explain brightness changes solely by motion [43, 81] or rely on illumination invariant features to cope with non-motion induced brightness changes [53, 121, 132, 189]. The data term we consider for our baseline, however, explicitly models illumination changes and hence enables the model to explain brightness variations in terms of illumination changes, which is why we refer to it as *illumination-aware* data term, cf. [8]. This brings the advantage of enabling the model to handle illumination changes without discarding potential useful information for the sake of robustness [54]. Hence, the data term of our baseline model is composed of an illumination compensated brightness constancy assumption and illumination compensated gradient constancy assumption, given by

$$\begin{aligned} D(\mathbf{w}, \mathbf{c}) = \Psi_{\mathrm{c}}\Big(\big(I_2(\mathbf{x}+\mathbf{w}) - \Phi(I_1(\mathbf{x}), \mathbf{c})\big)^2 \Big) \\ + \gamma \cdot \Psi_{\mathrm{c}}\Big(\big|\nabla I_2(\mathbf{x}+\mathbf{w}) - \nabla \Phi(I_1(\mathbf{x}), \mathbf{c})\big|^2 \Big) , \end{aligned} \tag{4.2}$$

where Ψ_{c} is the Charbonnier penalizer described in Section 2.4.3 of the foundation chapter, γ is a weighting parameter, and $\Phi(I, \mathbf{c})$ is a parametrized brightness transfer function [72]. In this context one may note that the use of additional illumination invariant features, i.e., the gradient constancy, helps to guide the estimation process. In practice this guidance helps to improve the estimation accuracy compared to a variant solely based on the illumination compensated brightness constancy assumption. The parametrized brightness transfer function $\Phi(I, \mathbf{c})$ maps the intensities of the first frame I_1 to the corresponding intensities of the second frame I_2. It is defined by the spatially varying coefficient fields $\mathbf{c}$, a given set of n basis functions $\phi_i : \mathbb{R} \to \mathbb{R}$, and the mean brightness transfer function $\bar{\phi}(I) : \mathbb{R} \to \mathbb{R}$. It reads

$$\Phi(I, \mathbf{c}) = \bar{\phi}(I) + \sum_{i=1}^{n} c_i \cdot \phi_i(I) \, . \tag{4.3}$$

In contrast to Demetz et al. [54] we do not learn the parametrized brightness transfer function from training data. Instead, we use a normalized affine brightness transfer function, which is defined by the following mean and basis functions

$$\bar{\phi}(I) = I \, , \quad \phi_1(I) = \frac{I}{n_1} \, , \quad \text{and} \quad \phi_2(I) = \frac{1}{n_2} \, , \tag{4.4}$$

where n_1 and n_2 are normalization factors such that $|\phi_i(I)| = 1$. Although such a normalized affine basis function might not offer an ideal representation for a specific domain, it allows to model most of the occurring illumination changes while offering an intuitive interpretation of the coefficient fields compared to a learned basis function as used in [54].

REGULARIZATION TERMS In general, the data term on its own does not provide enough information to obtain either a unique or a satisfying solution for the entire image domain, hence spatial regularization of the unknowns is required. In contrast to classical approaches that do not explicitly model illumination changes, this becomes even more important in our baseline, because observed brightness changes can be explained in multiple ways, i.e., in terms of motion or in terms of illumination changes.

In case of the coefficient regularizer R_{illum}, our baseline uses an anisotropic first-order regularizer [216], which models the assumption that neighboring locations are exposed to similar illumination changes and therefore can be expressed by piecewise constant coefficient fields $\mathbf{c}$. Furthermore, it assumes that discontinuities in the coefficient fields align with image edges in the uncompensated reference frame I_1. This makes sense, since illumination changes tend to align with these edges, e.g., in case of shadow edges. The corresponding regularizer is given by

$$R_{\text{illum}}(\mathbf{c}) = \sum_{l=1}^{2} \Psi_l \left(\sum_{i=1}^{n} \left(\mathbf{r}_l^\top \nabla c_i \right)^2 \right), \tag{4.5}$$

where $\mathbf{r}_1$ and $\mathbf{r}_2$ denote two spatially varying orthogonal directions, i.e., orthonormal vectors, that enable the desired direction depended smoothing behavior. In practice, these directions are extracted as the eigenvectors of either the structure tensor [59] or the regularization tensor [216] and typically represent directions across and along image edges. In our baseline, the penalizer functions Ψ_1 and Ψ_2 corresponding to the directions $\mathbf{r}_1$ and $\mathbf{r}_2$ are chosen to be the edge-enhancing Perona-Malik penalizer and the edge-preserving Charbonnier penalizer, respectively, what allows to capture sharp discontinuities, see Section 2.4.3.

While classical as well as many recent variational optical flow methods resort to such first-order regularizers also in case of the flow regularizer R_{flow}, the model of Demetz et al. [54] employs a second-order regularization strategy which directly penalizes the second-order derivatives of the unknowns. Such a second-order regularizer has the advantage that it does not favor piecewise constant solutions but piecewise affine solutions, which makes it more suitable for scenarios that contain a large amount of non-fronto parallel motion. The corresponding regularizer reads

$$R_{\text{flow}}(\mathbf{w}) = \Psi_c \left(|\mathcal{H}u|_{\text{F}}^2 + |\mathcal{H}v|_{\text{F}}^2 \right), \tag{4.6}$$

where $|\cdot|_{\text{F}}$ is the Frobenius norm, and $\mathcal{H}u$ and $\mathcal{H}v$ are the Hessians of u and v, respectively. As penalizer function the edge-preserving Charbonnier penalizer Ψ_c is used, see Section 2.4.3.

4.1.4 Regularizers

After introducing our baseline optical flow model, we turn to the prototypes of the three modeling strategies, i.e., direct, combined and indirect approaches. Furthermore, we introduce isotropic and anisotropic variants for each of the prototypes. Hence, for the sake of completeness, let us start with a first-order regularizer.

4.1.4.1 First-Order Regularization

Isotropic Since the seminal work of Horn and Schunck [81] first-order regularizers have a long successful tradition within optical flow estimation. Even quite recent works still use such regularizers, see Revaud et al. [143]. A well-known isotropic first-order regularizer is given by [36]:

$$R_{\text{1-iso}}(\mathbf{w}) = \Psi \left(|\nabla u|^2 + |\nabla v|^2 \right), \tag{4.7}$$

where Ψ is a penalizer function that allows preserving discontinuities in the flow field. This type of smoothness term also comprises some variants of the well-known total variation regularization (TV) [147, 210].

ANISOTROPIC To improve the performance at object boundaries, i.e., by smoothing along object edges but not across them, researchers proposed several anisotropic extensions. Going back to the work of Nagel and Enkelmann [124], the idea is to exploit directional information to steer the smoothing [162, 185, 193, 216]. To derive the anisotropic counterpart of the isotropic prototype in Equation 4.7, it can be rewritten using the unit vectors $\mathbf{e}_1 = (1,0)^\top$ and $\mathbf{e}_2 = (0,1)^\top$:

$$R_{\text{1-iso}}(\mathbf{w}) = \Psi\left(\sum_{l=1}^{2}\left(\mathbf{e}_l^\top \nabla u\right)^2 + \left(\mathbf{e}_l^\top \nabla v\right)^2\right). \tag{4.8}$$

Here, $\mathbf{e}_l^\top \nabla u$ and $\mathbf{e}_l^\top \nabla v$ are the directional derivatives $\partial_{\mathbf{e}_l} u$ and $\partial_{\mathbf{e}_l} v$, respectively. By replacing the unit vectors $\mathbf{e}_1, \mathbf{e}_2$ with locally varying directions $\mathbf{r}_1, \mathbf{r}_2$, which form an orthonormal basis, and by applying the penalization to both directions separately, one obtains the anisotropic counterpart of Equation 4.7, given by [162, 216]:

$$R_{\text{1-aniso}}(\mathbf{w}) = \sum_{l=1}^{2}\Psi_l\left(\left(\mathbf{r}_l^\top \nabla u\right)^2 + \left(\mathbf{r}_l^\top \nabla v\right)^2\right). \tag{4.9}$$

Please note that this model comprises the complementary regularizer from Zimmer et al. [216], which is related to the steered random field model of Sun et al. [162]. For determining the local directions $\mathbf{r}_1, \mathbf{r}_2$, one can use the eigenvectors of either the structure tensor [162] or the regularization tensor [216].

4.1.4.2 DIRECT SECOND-ORDER REGULARIZATION

ISOTROPIC Probably the most intuitive way to model second-order smoothness assumptions is to penalize the second-order derivatives of the flow directly. A prominent isotropic example is based on the Hessian, which already has been introduced together with our baseline model proposed by Demetz et al. [54]. It is given by

$$R_{\text{2-iso}}(\mathbf{w}) = \Psi\left(|\mathcal{H}u|_F^2 + |\mathcal{H}v|_F^2\right), \tag{4.10}$$

where $|\cdot|_F$ is the Frobenius norm, and $\mathcal{H}u$ and $\mathcal{H}v$ is the Hessian of u and v, respectively. This regularizer has also been applied in the context of denoising [115, 149] and Shape from Shading [94, 177]. Other matrix norms generalizing the Frobenius norm, e.g., the l_p-norm [149] ($p = 1$) or the Schatten$_p$-norm [107] ($p = 1, 2, \infty$), have been considered in the literature as well.

ANISOTROPIC To derive the corresponding anisotropic counterpart as for the first-order case, we reformulate the isotropic regularizer using the unit vectors $\mathbf{e}_1, \mathbf{e}_2$:

$$R_{\text{2-iso}}(\mathbf{w}) = \Psi\left(\sum_{l=1}^{2}\sum_{m=1}^{2}\left(\mathbf{e}_m^\top \mathcal{H}u\, \mathbf{e}_l\right)^2 + \left(\mathbf{e}_m^\top \mathcal{H}v\, \mathbf{e}_l\right)^2\right), \tag{4.11}$$

where $\mathbf{e}_m^\top \mathcal{H} u \, \mathbf{e}_l$ and $\mathbf{e}_m^\top \mathcal{H} v \, \mathbf{e}_l$ are the second-order directional derivatives $\partial_{\mathbf{e}_m \mathbf{e}_l} u$ and $\partial_{\mathbf{e}_m \mathbf{e}_l} v$, respectively. Once again the unit vectors $\mathbf{e}_1, \mathbf{e}_2$ are replaced with the locally varying directions $\mathbf{r}_1$, $\mathbf{r}_2$. As observed by Lellmann et al. [108] in the context of denoising, two possibilities arise how to penalize the directional derivatives. One can either penalize the directions $\mathbf{r}_k$ jointly or penalize all the directional derivatives separately. Please note that this constitutes a substantial difference to the first-order case, where both options coincide. For our second-order term in Equation 4.11 the first case leads to

$$\begin{aligned} R_{\text{2-aniso-single}}(\mathbf{w}) &= \sum_{l=1}^{2} \Psi_l \left(\sum_{m=1}^{2} \left(\mathbf{r}_m^\top \mathcal{H} u \, \mathbf{r}_l \right)^2 + \left(\mathbf{r}_m^\top \mathcal{H} v \, \mathbf{r}_l \right)^2 \right) \\ &= \sum_{l=1}^{2} \Psi_l \left(|\mathcal{H} u \, \mathbf{r}_l|^2 + |\mathcal{H} v \, \mathbf{r}_l|^2 \right), \end{aligned} \tag{4.12}$$

which we refer to as *single anisotropic* regularization. The latter case yields

$$R_{\text{2-aniso-double}}(\mathbf{w}) = \sum_{l=1}^{2} \sum_{m=1}^{2} \Psi_{l,m} \left(\left(\mathbf{r}_m^\top \mathcal{H} u \, \mathbf{r}_l \right)^2 + \left(\mathbf{r}_m^\top \mathcal{H} v \, \mathbf{r}_l \right)^2 \right), \tag{4.13}$$

which we refer to as *double anisotropic* regularization. To the best of our knowledge, we are the first to use both the single and the double anisotropic variant in the context of motion estimation.

4.1.4.3 Combined Regularization

Isotropic As a representative of the class of combined regularizers we consider an infimal convolution approach. To this end, we additively split the actual flow field $\mathbf{w}$ into two individual components $\mathbf{w}_i = (u_i, v_i)^\top$ with $i = \{1, 2\}$ such that $\mathbf{w} = \mathbf{w}_1 + \mathbf{w}_2$. Furthermore, a direct first and direct second-order regularizer are combined by applying them individually to the two flow components and by balancing them via the parameter λ. The resulting isotropic variant reads

$$R_{\text{inf-iso}}(\mathbf{w}) = \inf_{\mathbf{w}=\mathbf{w}_1+\mathbf{w}_2} \left\{ R_{\text{1-iso}}(\mathbf{w}_1) + \lambda \cdot R_{\text{2-iso}}(\mathbf{w}_2) \right\}. \tag{4.14}$$

Anisotropic Consequently, combining the single anisotropic variants of the direct first-order and second-order regularizers from Equation 4.9 and Equation 4.12 yields

$$R_{\text{inf-aniso-single}}(\mathbf{w}) = \inf_{\mathbf{w}=\mathbf{w}_1+\mathbf{w}_2} \left\{ R_{\text{1-aniso}}(\mathbf{w}_1) + \lambda \cdot R_{\text{2-aniso-single}}(\mathbf{w}_2) \right\}, \tag{4.15}$$

which resembles the combined approach for scalar-valued denoising from Lenzen et al. [109]. Analogously, using the double anisotropic counterparts, we obtain the double anisotropic variant

$$R_{\text{inf-aniso-double}}(\mathbf{w}) = \inf_{\mathbf{w}=\mathbf{w}_1+\mathbf{w}_2} \left\{ R_{\text{1-aniso}}(\mathbf{w}_1) + \lambda \cdot R_{\text{2-aniso-double}}(\mathbf{w}_2) \right\}. \tag{4.16}$$

As before, we are not aware of any motion estimation methods where such anisotropic infimal convolution regularizers have been applied.

4.1.4.4 Indirect Second-Order Regularization

Isotropic Finally, as a representative of the class of indirect approaches, we consider a coupling approach. It consists of two terms: a coupling term that models the similarity of the gradients to auxiliary functions and a smoothness term that enforces smoothness on these auxiliary functions. An isotropic variant is given by [33]:

$$R_{\text{c-iso}}(\mathbf{w}) = \inf_{\mathbf{a},\mathbf{b}} \left\{ C_{\text{c-iso}}(\mathbf{w},\mathbf{a},\mathbf{b}) + \lambda \cdot S_{\text{c-iso}}(\mathbf{a},\mathbf{b}) \right\}, \tag{4.17}$$

where $C_{\text{c-iso}}$ denotes the coupling term and $S_{\text{c-iso}}$ the smoothness term given by

$$C_{\text{c-iso}}(\mathbf{w},\mathbf{a},\mathbf{b}) = \Psi\left(|\nabla u - \mathbf{a}|^2 + |\nabla v - \mathbf{b}|^2 \right), \tag{4.18}$$

$$S_{\text{c-iso}}(\mathbf{a},\mathbf{b}) = \Psi\left(|\mathcal{J}\mathbf{a}|_F^2 + |\mathcal{J}\mathbf{b}|_F^2 \right), \tag{4.19}$$

respectively. Here, $\mathbf{a} = (a_1, a_2)^\top$ and $\mathbf{b} = (b_1, b_2)^\top$ are the auxiliary vector fields that approximate the gradients ∇u and ∇v, respectively, $\mathcal{J}\mathbf{a}$ and $\mathcal{J}\mathbf{b}$ denote the Jacobians of $\mathbf{a}$ and $\mathbf{b}$, and λ serves as weighting parameter. This type of smoothness term comprises the well-known total generalized variation regularizer (TGV) [34].

Anisotropic By first rewriting the isotropic case using the unit vectors $\mathbf{e}_1, \mathbf{e}_2$ as

$$C_{\text{c-iso}}(\mathbf{w},\mathbf{a},\mathbf{b}) = \Psi\left(\sum_{l=1}^{2} \left(\mathbf{e}_l^\top (\nabla u - \mathbf{a}) \right)^2 + \left(\mathbf{e}_l^\top (\nabla v - \mathbf{b}) \right)^2 \right), \tag{4.20}$$

$$S_{\text{c-iso}}(\mathbf{a},\mathbf{b}) = \Psi\left(\sum_{l=1}^{2} \sum_{m=1}^{2} \left(\mathbf{e}_m^\top \mathcal{J}\mathbf{a}\, \mathbf{e}_l \right)^2 + \left(\mathbf{e}_m^\top \mathcal{J}\mathbf{b}\, \mathbf{e}_l \right)^2 \right), \tag{4.21}$$

and then introducing the directions $\mathbf{r}_1, \mathbf{r}_2$ with separate penalization we obtain the single anisotropic case which resembles a vector-valued extension of the anisotropic coupling model of Hafner et al. [75] that we used in Chapter 3 for our 3D reconstruction method and was originally proposed in the context of focus fusion. It reads

$$R_{\text{c-aniso-single}}(\mathbf{w}) = \inf_{\mathbf{a},\mathbf{b}} \left\{ C_{\text{c-aniso-single}}(\mathbf{w},\mathbf{a},\mathbf{b}) + \lambda \cdot S_{\text{c-aniso-single}}(\mathbf{a},\mathbf{b}) \right\}, \tag{4.22}$$

with the respective coupling and smoothness terms

$$C_{\text{c-aniso}}(\mathbf{w},\mathbf{a},\mathbf{b}) = \sum_{l=1}^{2} \Psi_l\left(\left(\mathbf{r}_l^\top (\nabla u - \mathbf{a}) \right)^2 + \left(\mathbf{r}_l^\top (\nabla v - \mathbf{b}) \right)^2 \right), \tag{4.23}$$

$$\begin{aligned} S_{\text{c-aniso-single}}(\mathbf{a},\mathbf{b}) &= \sum_{l=1}^{2} \Psi\left(\sum_{m=1}^{2} \left(\mathbf{r}_m^\top \mathcal{J}\mathbf{a}\, \mathbf{r}_l \right)^2 + \left(\mathbf{r}_m^\top \mathcal{J}\mathbf{b}\, \mathbf{r}_l \right)^2 \right) \\ &= \sum_{l=1}^{2} \Psi\left(|\mathcal{J}\mathbf{a}\, \mathbf{r}_l|^2 + |\mathcal{J}\mathbf{b}\, \mathbf{r}_l|^2 \right). \end{aligned} \tag{4.24}$$

Again penalizing all directions separately results in the corresponding double anisotropic variant:

$$R_{\text{c-aniso-double}}(\mathbf{w}) = \inf_{\mathbf{a},\mathbf{b}} \left\{ C_{\text{c-aniso-double}}(\mathbf{w}, \mathbf{a}, \mathbf{b}) + \lambda \cdot S_{\text{c-aniso-double}}(\mathbf{a}, \mathbf{b}) \right\}, \tag{4.25}$$

with the respective coupling and smoothness terms given by

$$C_{\text{c-aniso}}(\mathbf{w}, \mathbf{a}, \mathbf{b}) = \sum_{l=1}^{2} \Psi_l \left(\left(\mathbf{r}_l^\top (\nabla u - \mathbf{a}) \right)^2 + \left(\mathbf{r}_l^\top (\nabla v - \mathbf{b}) \right)^2 \right), \tag{4.26}$$

$$S_{\text{c-aniso-double}}(\mathbf{a}, \mathbf{b}) = \sum_{l=1}^{2} \sum_{m=1}^{2} \Psi_{l,m} \left(\left(\mathbf{r}_m^\top \mathcal{J}\mathbf{a}\, \mathbf{r}_l \right)^2 + \left(\mathbf{r}_m^\top \mathcal{J}\mathbf{b}\, \mathbf{r}_l \right)^2 \right), \tag{4.27}$$

where $\mathbf{r}_m^\top \mathcal{J}\mathbf{a}\, \mathbf{r}_l$ and $\mathbf{r}_m^\top \mathcal{J}\mathbf{b}\, \mathbf{r}_l$ can be considered to be an approximation of the directional derivative $\mathbf{r}_k^\top \mathcal{H}u\, \mathbf{r}_l$ and $\mathbf{r}_k^\top \mathcal{H}v\, \mathbf{r}_l$, respectively. Also for the anisotropic coupling models, we are not aware of any motion estimation method that makes use of such regularizers. Regarding the anisotropic coupling term, the work of Ranftl [136] is close in spirit, which introduces anisotropic concepts in the coupling term but not the smoothness term.

4.1.5 Diffusion Processes

So far we not only juxtaposed different existing regularizers but also introduced new regularizers for the three different modeling strategies. Next, we want to analyze these regularizers. Therefore, we derive the gradient descent equations w.r.t. the unknowns of the respective regularizers to reveal the underlying diffusion process [188]. These resulting diffusion processes differ in the order of the involved derivatives as well as in the degree of anisotropy. Analyzing these diffusion processes allows us to gain a better understanding of the anisotropy and to highlight the commonalities between the different techniques. To simplify the analysis, we first provide a summary of the general structure of the underlying diffusion processes.

4.1.5.1 Second-Order Diffusion

In case of the standard first-order regularization (see Equation 4.7, Equation 4.9, Equation 4.14, Equation 4.15, and Equation 4.16) two coupled scalar-valued non-linear second-order diffusion processes occur. We can write the associated scalar-valued diffusion equations as

$$\partial_t u = \nabla \cdot (T_1 \nabla u), \tag{4.28}$$

$$\partial_t v = \nabla \cdot (T_1 \nabla v), \tag{4.29}$$

where ∂_t denotes an artificial time derivative, $\nabla \cdot$ the divergence operator, and T_1 is the well-known 2×2 symmetric positive-definite diffusion tensor that describes the diffusion process [187], which has the following structure:

$$T_1 = \begin{pmatrix} a & b \\ b & c \end{pmatrix}, \tag{4.30}$$

where the entries a, b and c are scalars. Furthermore, we introduce $\boldsymbol{\nabla}_n = I_{n\times n} \otimes \nabla$, a generalization of the classical nabla operator $\nabla = \boldsymbol{\nabla}_1$, where $I_{n\times n}$ denotes the $n \times n$ identity matrix, $\otimes$ is the Kronecker product and the resulting dimension is given by $2n \times n$. This operator allows rewriting the coupled diffusion process in terms of a vector-valued diffusion of the flow $\mathbf{w} = (u, v)^\top$:

$$\partial_t \mathbf{w} = \boldsymbol{\nabla}_2 \cdot (\mathbf{T}_1 \, \boldsymbol{\nabla}_2 \mathbf{w}) \,, \tag{4.31}$$

where $\mathbf{T}_1$ is a symmetric positive definite 4×4 tensor with a block diagonal structure holding the actual diffusion tensor

$$\mathbf{T}_1 = I_{2\times 2} \otimes T_1 = \begin{pmatrix} T_1 & 0 \\ 0 & T_1 \end{pmatrix} . \tag{4.32}$$

This generalized notation offers a convenient way to formulate vector-valued second-order diffusion processes. In particular, it allows to intuitively spot relations to special cases of second-order diffusion, e.g., guided diffusion and generalized coupled diffusion, which we detail next.

Guided Diffusion One special case of second-order diffusion arises in the context of the indirect second-order regularization, i.e., guided diffusion [133]. In particular, in case of the flow components the involved coupling term (see Equation 4.18, Equation 4.23, and Equation 4.26) leads to a guided second-order diffusion process, for which we can write the associated coupled diffusion equations as

$$\partial_t u = \nabla \cdot (T_1^{\text{aux}} (\nabla u - \mathbf{a})) \,, \tag{4.33}$$
$$\partial_t v = \nabla \cdot (T_1^{\text{aux}} (\nabla v - \mathbf{b})) \,, \tag{4.34}$$

where T_1^{aux} is as before the 2×2 symmetric positive-definite diffusion tensor and the vector-valued auxiliary functions $\mathbf{a}$ and $\mathbf{b}$ act as guidance functions, respectively. Again, we can write the diffusion process more compactly using a vector-valued notation

$$\partial_t \mathbf{w} = \boldsymbol{\nabla}_2 \cdot \left(\mathbf{T}_1^{\text{aux}} \left(\boldsymbol{\nabla}_2 \mathbf{w} - \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} \right) \right) , \tag{4.35}$$

where $\mathbf{T}_1^{\text{aux}}$ is again the stacked diffusion tensor

$$\mathbf{T}_1^{\text{aux}} = I_{2\times 2} \otimes T_1^{\text{aux}} = \begin{pmatrix} T_1^{\text{aux}} & 0 \\ 0 & T_1^{\text{aux}} \end{pmatrix} . \tag{4.36}$$

Here, one can see that the standard vector valued-diffusion, i.e., Equation 4.31, as well as the guided vector valued-diffusion, i.e., Equation 4.35, are solely coupled by T_1 and T_1^{aux}, respectively. Furthermore, note that for $\mathbf{a} = \mathbf{b} = \mathbf{0}$, the structure of Equation 4.35 comes down to the structure of the standard second diffusion given in Equation 4.31.

Generalized Coupled Diffusion Another special case is given by the generalized coupled diffusion process that may occur in case of the indirect second-order. To be more precise, the smoothness term of the auxiliary functions $\mathbf{a} = (a_1, a_2)^\top$ and $\mathbf{b} = (b_1, b_2)^\top$ (see Equa-

tion 4.19, Equation 4.24, and Equation 4.27) leads to two coupled vector-valued second-order diffusion processes, for which the scalar-valued diffusion equations can be written as

$$\partial_t a_1 = \nabla \cdot (T_A \nabla a_1) + \nabla \cdot (T_B \nabla a_2)\,, \tag{4.37}$$
$$\partial_t a_2 = \nabla \cdot (T_B \nabla a_1) + \nabla \cdot (T_C \nabla a_2)\,, \tag{4.38}$$

and

$$\partial_t b_1 = \nabla \cdot (T_A \nabla b_1) + \nabla \cdot (T_B \nabla b_2)\,, \tag{4.39}$$
$$\partial_t b_2 = \nabla \cdot (T_B \nabla b_1) + \nabla \cdot (T_C \nabla b_2)\,, \tag{4.40}$$

where T_A, T_B and T_C denote the respective 2×2 positive definite second-order diffusion tensors. We refer to all the cases where T_B is not a zero matrix as generalized coupled (Equation 4.27), since in this case the interaction is not only given through the diffusion tensor (i.e., T_A or T_C) but also by additional contributions by another unknown, e.g., $\nabla \cdot (T_B \nabla a_2)$. Introducing a 4×4 matrix composed of these three possibly different 2×2 diffusion tensors

$$T_2^{\text{aux}} = \begin{pmatrix} T_A & T_B \\ T_B & T_C \end{pmatrix}, \tag{4.41}$$

allows writing the diffusion process compactly in a vector formulation as

$$\partial_t \mathbf{a} = \boldsymbol{\nabla}_2 \cdot (T_2^{\text{aux}} \boldsymbol{\nabla}_2 \mathbf{a})\,, \tag{4.42}$$
$$\partial_t \mathbf{b} = \boldsymbol{\nabla}_2 \cdot (T_2^{\text{aux}} \boldsymbol{\nabla}_2 \mathbf{b})\,. \tag{4.43}$$

This diffusion process, given by two vector-valued diffusion equations, can be seen as an vector-valued extension to the standard second-order diffusion, e.g., given by two scalar-valued diffusion equations as shown in Equation 4.28 and Equation 4.29. While in the standard second-order diffusion case the diffusion equations are coupled via the diffusion tensor T_1, the diffusion equations in the generalized coupled diffusion process are coupled via the analogous counterpart T_2^{aux}.

4.1.5.2 Fourth-Order Diffusion

Finally, in case of the direct second-order regularization (see Equation 4.10, Equation 4.12, and Equation 4.13) and the combined regularization (see Equation 4.14, Equation 4.15 and Equation 4.16), a coupled non-linear fourth-order diffusion process takes place. Here, we can write the corresponding diffusion equations for the flow components in a similar fashion. Therefore, we introduce $\nabla^2 = \nabla \otimes \nabla = (\partial_{xx}, \partial_{xy}, \partial_{yx}, \partial_{yy})^\top$ as a kind of second-order nabla operator and analog $\nabla^2 \cdot$ as a second-order divergence equivalent. These operators allow us to write the scalar-valued diffusion equations as

$$\partial_t u = \nabla^2 \cdot \left(T_2\, \nabla^2 u\right), \tag{4.44}$$
$$\partial_t v = \nabla^2 \cdot \left(T_2\, \nabla^2 v\right), \tag{4.45}$$

where T_2 is the 4×4 fourth-order diffusion tensor with the following structure

$$T_2 = \begin{pmatrix} A & B \\ B & C \end{pmatrix}, \tag{4.46}$$

where A, B and C are 2×2 matrices. A derivation hereof can be found in Section A.2. As in the second-order diffusion case, we can further generalize the differential operator ∇^2 via $\boldsymbol{\nabla}_n^2 = I_{n\times n} \otimes \nabla^2$. This allows to state the vector-valued formulation:

$$\partial_t \mathbf{w} = \boldsymbol{\nabla}_2^2 \cdot \left(\mathbf{T}_2 \, \boldsymbol{\nabla}_2^2 \mathbf{w} \right), \tag{4.47}$$

where $\mathbf{T}_2$ is a 8×8 matrix with a block diagonal structure holding the actual diffusion tensor

$$\mathbf{T}_2 = I_{2\times 2} \otimes T_2 = \begin{pmatrix} T_2 & 0 \\ 0 & T_2 \end{pmatrix}. \tag{4.48}$$

4.1.6 Diffusion Tensors

In the previous section, we reviewed the general form of the occurring diffusion processes. In all cases, the underlying diffusion tensor guides the diffusion process. More precisely, the diffusion tensor encodes the anisotropy of the diffusion process: the smoothing direction is determined by the eigenvectors and the magnitude by the eigenvalues. Next, we analyze the respective diffusion tensor that corresponds to each of the introduced regularizers.

4.1.6.1 First-Order Regularization

The first-order regularization induces a second-order diffusion process. This process involves a 2×2 diffusion tensor as shown in Equation 4.30.

Isotropic In the isotropic case of the first-order regularizer, the diffusion tensor is given by

$$\begin{aligned} T_{\text{1-iso}} &= \sum_{l=1}^{2} \Psi'\Big(|\nabla u|^2 + |\nabla v|^2 \Big) \cdot \mathbf{e}_l \, \mathbf{e}_l^\top \\ &= I_{2\times 2} \cdot \Psi'\Big(|\nabla u|^2 + |\nabla v|^2 \Big), \end{aligned} \tag{4.49}$$

which makes explicit that no directional-dependent smoothing occurs since the only eigenvalue is twofold. Assuming Ψ' is positive and decreasing, what holds for all the considered penalizer functions in this thesis, the smoothing is reduced if the derivatives are large, which typically corresponds to a discontinuity within the unknowns.

Anisotropic In the anisotropic case the diffusion tensor reads

$$T_{\text{1-aniso}} = \sum_{l=1}^{2} \Psi_l'\Big(\left(\mathbf{r}_l^\top \nabla u \right)^2 + \left(\mathbf{r}_l^\top \nabla v \right)^2 \Big) \cdot \mathbf{r}_l \, \mathbf{r}_l^\top. \tag{4.50}$$

Here one can see that the two eigenvalues may differ, which allows a directional-dependent smoothing. The smoothing along a specific direction $\mathbf{r}_l$ is reduced if the corresponding directional derivative is large, but in contrast to the isotropic case, this does not necessarily affect the orthogonal direction. Consequently, the separate penalization allows adapting the smoothing behavior to homogeneous regions, edges, and corners.

4.1.6.2 Direct Second-Order Regularization

The direct second-order regularization induces a fourth-order diffusion process. This process involves a 4×4 diffusion tensor as shown in Equation 4.46.

Isotropic In the isotropic variant of the direct second-order regularization, where we have a fourth-order diffusion process, the diffusion tensor reads

$$\begin{aligned} T_{\text{2-iso}} &= \sum_{l=1}^{2} \sum_{m=1}^{2} \Psi'\Big(|\mathcal{H}u|_F^2 + |\mathcal{H}v|_F^2 \Big) \cdot \Big(\mathbf{e}_m \mathbf{e}_m^\top \otimes \mathbf{e}_l \mathbf{e}_l^\top \Big) \\ &= I_{4\times 4} \cdot \Psi'\Big(|\mathcal{H}u|_F^2 + |\mathcal{H}v|_F^2 \Big) . \end{aligned} \tag{4.51}$$

As in the first-order regularization scenario, this makes explicit that no directional adaptation takes place since again the only eigenvalue is fourfold.

Anisotropic In the single anisotropic case, the fourth-order diffusion tensor reads

$$\begin{aligned} T_{\text{2-aniso-s}} &= \sum_{l=1}^{2} \sum_{m=1}^{2} \Psi_l'\left(\sum_{m=1}^{2} \Big(\mathbf{e}_m^\top \mathcal{H}u\, \mathbf{r}_l \Big)^2 + \Big(\mathbf{e}_m^\top \mathcal{H}v\, \mathbf{r}_l \Big)^2 \right) \cdot \Big(\mathbf{e}_m \mathbf{e}_m^\top \otimes \mathbf{r}_l \mathbf{r}_l^\top \Big) \\ &= I_{2\times 2} \otimes \left(\sum_{l=1}^{2} \Psi_l'\Big(|\mathcal{H}u\, \mathbf{r}_l|^2 + |\mathcal{H}v\, \mathbf{r}_l|^2 \Big) \cdot \mathbf{r}_l \mathbf{r}_l^\top \right) , \end{aligned} \tag{4.52}$$

where there are two eigenvalues Ψ_l' which are twofold. By rewriting this equation we can make the block diagonal structure of the fourth-order diffusion tensor in the single anisotropic case explicit

$$T_{\text{2-aniso-s}} = \begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix}, \quad \text{with} \quad A = \sum_{l=1}^{2} \Psi_l'\Big(|\mathcal{H}u\, \mathbf{r}_l|^2 + |\mathcal{H}v\, \mathbf{r}_l|^2 \Big) \cdot \mathbf{r}_l \mathbf{r}_l^\top . \tag{4.53}$$

We provide a derivation of this block structure in Section A.2 of the appendix. In the double anisotropic case, the respective diffusion tensor reads

$$T_{\text{2-aniso-d}} = \sum_{l=1}^{2} \sum_{m=1}^{2} \Psi_{l,m}'\left(\Big(\mathbf{r}_m^\top \mathcal{H}u\, \mathbf{r}_l \Big)^2 + \Big(\mathbf{r}_m^\top \mathcal{H}v\, \mathbf{r}_l \Big)^2 \right) \cdot \Big(\mathbf{r}_m \mathbf{r}_m^\top \otimes \mathbf{r}_l \mathbf{r}_l^\top \Big) , \tag{4.54}$$

where all four eigenvalues may differ, which constitutes maximal adaptation.

4.1.6.3 Combined Regularization

In case of the combined regularizer, the same diffusion tensors arise as for the first-order and the direct second-order case. In contrast, however, both diffusion tensors emerge together in the minimization process. Although we cannot obtain any additional insights from the actual tensors, we specify them briefly for the sake of completeness.

Isotropic The isotropic case leads to the following two diffusion tensors:

$$T_{\text{1-inf-iso}} = I_{2\times 2} \cdot \Psi'\Big(|\nabla u_1|^2 + |\nabla v_1|^2\Big)\,, \tag{4.55}$$

$$T_{\text{2-inf-iso}} = I_{4\times 4} \cdot \Psi'\Big(|\mathcal{H} u_2|_F^2 + |\mathcal{H} v_2|_F^2\Big)\,. \tag{4.56}$$

Anisotropic The anisotropic variants result in the following diffusion tensors:

$$T_{\text{1-inf-aniso}} = \sum_{l=1}^{2} \Psi_l'\Big(\Big(\mathbf{r}_l^\top \nabla u_1\Big)^2 + \Big(\mathbf{r}_l^\top \nabla v_1\Big)^2\Big) \cdot \mathbf{r}_l\,\mathbf{r}_l^\top\,, \tag{4.57}$$

$$T_{\text{2-inf-aniso-s}} = I_{2\times 2} \otimes \left(\sum_{l=1}^{2} \Psi_l'\Big(|\mathcal{H} u_2\,\mathbf{r}_l|^2 + |\mathcal{H} v_2\,\mathbf{r}_l|^2\Big) \cdot \mathbf{r}_l \mathbf{r}_l^\top\right), \tag{4.58}$$

$$T_{\text{2-inf-aniso-d}} = \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi_{l,m}'\Big(\Big(\mathbf{r}_m^\top \mathcal{H} u_2\,\mathbf{r}_l\Big)^2 + \Big(\mathbf{r}_m^\top \mathcal{H} v_2\,\mathbf{r}_l\Big)^2\Big) \cdot \Big(\mathbf{r}_m \mathbf{r}_m^\top \otimes \mathbf{r}_l \mathbf{r}_l^\top\Big)\,. \tag{4.59}$$

4.1.6.4 Indirect Second-Order Regularization

The indirect second-order regularizer induces not only a guided diffusion process but also a generalized coupled diffusion process. Hence, we specify the involved 2×2 diffusion tensor as well as the stacked 4×4 diffusion tensor as specified in Equation 4.30 and Equation 4.41, respectively.

Isotropic The indirect second-order regularization involves multiple diffusion tensors again. In the isotropic case, the tensors are given by the 2×2 tensor resulting from the coupling term

$$\begin{aligned} T_{\text{1-iso}}^{\text{aux}} &= \sum_{l=1}^{2} \Psi'\Big(|\nabla u - \mathbf{a}|^2 + |\nabla v - \mathbf{b}|^2\Big) \cdot \mathbf{e}_l\,\mathbf{e}_l^\top \\ &= I_{2\times 2} \cdot \Psi'\Big(|\nabla u - \mathbf{a}|^2 + |\nabla v - \mathbf{b}|^2\Big)\,, \end{aligned} \tag{4.60}$$

and the 4×4 tensor resulting from the smoothness term

$$\begin{aligned} T_{\text{2-iso}}^{\text{aux}} &= \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'\Big(|\mathcal{J}\mathbf{a}|_F^2 + |\mathcal{J}\mathbf{b}|_F^2\Big) \cdot \Big(\mathbf{e}_m\,\mathbf{e}_m^\top \otimes \mathbf{e}_l\,\mathbf{e}_l^\top\Big) \\ &= I_{4\times 4} \cdot \Psi'\Big(|\mathcal{J}\mathbf{a}|_F^2 + |\mathcal{J}\mathbf{b}|_F^2\Big)\,. \end{aligned} \tag{4.61}$$

For both tensors, the only eigenvalue is twofold and fourfold, respectively. Hence, there is no directional-dependent smoothing.

ANISOTROPIC The diffusion tensor of the anisotropic variant is the same for both the single anisotropic case as well as the double anisotropic case. It reads

$$T^{\text{aux}}_{\text{1-aniso}} = \sum_{l=1}^{2} \Psi'_l\left(\left(\mathbf{r}_l^\top(\nabla u - \mathbf{a})\right)^2 + \left(\mathbf{r}_l^\top(\nabla v - \mathbf{b})\right)^2\right) \cdot \mathbf{r}_l\,\mathbf{r}_l^\top . \tag{4.62}$$

In contrast, the tensor resulting from smoothness term differs in the two cases. In the single anisotropic case, the diffusion tensor reads

$$\begin{aligned} T^{\text{aux}}_{\text{2-aniso-s}} &= \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_l\left(\sum_{m=1}^{2}\left(\mathbf{e}_m^\top \mathcal{J}\mathbf{a}\,\mathbf{r}_l\right)^2 + \left(\mathbf{e}_m^\top \mathcal{J}\mathbf{b}\,\mathbf{r}_l\right)^2\right) \cdot \left(\mathbf{e}_m\mathbf{e}_m^\top \otimes \mathbf{r}_l\mathbf{r}_l^\top\right) \\ &= I_{2\times 2} \otimes \left(\sum_{l=1}^{2} \Psi'_l\left(|\mathcal{J}\mathbf{a}\,\mathbf{r}_l|^2 + |\mathcal{J}\mathbf{b}\,\mathbf{r}_l|^2\right) \cdot \mathbf{r}_l\mathbf{r}_l^\top\right) . \end{aligned} \tag{4.63}$$

As before only up to two different eigenvalues may appear. In the double anisotropic case, the diffusion tensor is given by

$$T^{\text{aux}}_{\text{2-aniso-d}} = \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m}\left(\left(\mathbf{r}_m^\top \mathcal{J}\mathbf{a}\,\mathbf{r}_l\right)^2 + \left(\mathbf{r}_m^\top \mathcal{J}\mathbf{b}\,\mathbf{r}_l\right)^2\right) \cdot \left(\mathbf{r}_m\mathbf{r}_m^\top \otimes \mathbf{r}_l\mathbf{r}_l^\top\right) . \tag{4.64}$$

where all four eigenvalues may differ.

4.1.7 MINIMIZATION

After analyzing the diffusion processes resulting from the respective regularization strategies, we discuss the minimization of the considered prototypes. Here prototypes refer to the different variants of the considered baseline model in Equation 4.1, where we replace the original flow regularizer R_{flow} with the different regularizers given in Subsection 4.1.4. To minimize the resulting non-convex energy functionals, we resort to the coarse-to-fine warping strategy as described in the foundation chapter, see Subsection 2.4.2. Therefore, we first derive the differential formulations.

4.1.7.1 DIFFERENTIAL FORMULATION

By splitting the unknowns of each resolution level k of the coarse-to-fine scheme into known intermediate solutions $\mathbf{w}^k$, $\mathbf{c}^k$ and unknown increments $\mathbf{dw}^k$, $\mathbf{dc}^k$, the solution of the next finer resolution level $k+1$ is defined by

$$\mathbf{w}^{k+1} = \mathbf{w}^k + \mathbf{dw}^k , \qquad \text{and} \qquad \mathbf{c}^{k+1} = \mathbf{c}^k + \mathbf{dc}^k . \tag{4.65}$$

Using this splitting, we can write the differential formulation of the baseline energy given in Equation 4.1 for the unknown increments as

$$E(\mathbf{dw}^k, \mathbf{dc}^k) = \int_\Omega D(\mathbf{dw}^k, \mathbf{dc}^k) + \beta \cdot R_{\text{illum}}(\mathbf{dc}^k) + \alpha \cdot R_{\text{flow}}(\mathbf{dw}^k)\, d\mathbf{x} , \tag{4.66}$$

where the different terms of the original energy functional are replaced by the corresponding differential formulations, which are specified next.

DATA TERM In the case of the data term, we linearize the original term w.r.t. the unknown increments. Since the term is already linear in the coefficient increments $\mathbf{dc}^k$ (the brightness transfer function Φ is a linear combination of weighted basis functions, cf. Equation 4.3), only the expressions $I_2(\mathbf{x}+\mathbf{w}^{k+1})$ in the brightness constancy assumption and $\nabla I_2(\mathbf{x}+\mathbf{w}^{k+1})$ in the gradient constancy assumption, have to be linearized. Applying a first-order Taylor expansion results in

$$I_2(\mathbf{x}+\mathbf{w}^{k+1}) \approx \nabla I_2(\mathbf{x}+\mathbf{w}^k)^\top \mathbf{dw}^k + I_2(\mathbf{x}+\mathbf{w}^k)\,, \tag{4.67}$$

$$\nabla I_2(\mathbf{x}+\mathbf{w}^{k+1}) \approx \mathcal{H}(I_2(\mathbf{x}+\mathbf{w}^k))^\top \mathbf{dw}^k + \nabla I_2(\mathbf{x}+\mathbf{w}^k)\,. \tag{4.68}$$

Introducing the abbreviations $I_1 := I_1(\mathbf{x})$ and $I_2^k := I_2(\mathbf{x}+\mathbf{w}^k)$ allows us now to write the non-normalized differential formulation of the data term as

$$\begin{aligned} D(\mathbf{dw}^k, \mathbf{dc}^k) = \Psi_{\mathrm{c}}\Big(\Big(\nabla I_2^{k\top}\mathbf{dw}^k + I_2^k - \Phi(I_1, \mathbf{c}^k + \mathbf{dc}^k)\Big)^2\Big) \\ + \gamma \cdot \Psi_{\mathrm{c}}\Big(\Big|\mathcal{H}(I_2^k)^\top \mathbf{dw}^k + \nabla I_2^k - \nabla\Phi(I_1, \mathbf{c}^k + \mathbf{dc}^k)\Big|^2\Big)\,. \end{aligned} \tag{4.69}$$

As in the previous chapter, we apply an additional constraint normalization [158, 216] to both constancy assumptions via the following normalization factors

$$\theta^k := \Big(|\nabla I_2^k|^2 + \zeta^2\Big)^{-\frac{1}{2}}\,, \tag{4.70}$$

$$\theta_x^k := \Big(|\nabla I_{2,x}^k|^2 + \zeta^2\Big)^{-\frac{1}{2}}\,, \quad \theta_y^k := \Big(|\nabla I_{2,y}^k|^2 + \zeta^2\Big)^{-\frac{1}{2}}\,, \tag{4.71}$$

where $\zeta = 0.01$ is a small parameter to prevent division by zero. Combining linearization and normalization, we obtain the normalized differential formulation as

$$\begin{aligned} D(\mathbf{dw}^k, \mathbf{dc}^k) = \Psi_{\mathrm{c}}\Big(\Big(\theta^k\Big(\nabla I_2^{k\top}\mathbf{dw}^k + I_2^k - \Phi(I_1, \mathbf{c}^k + \mathbf{dc}^k)\Big)\Big)^2\Big) \\ + \gamma \cdot \Psi_{\mathrm{c}}\Big(\Big|\theta_{xy}^k\Big(\mathcal{H}(I_2^k)^\top \mathbf{dw}^k + \nabla I_2^k - \nabla\Phi(I_1, \mathbf{c}^k + \mathbf{dc}^k)\Big)\Big|^2\Big)\,, \end{aligned} \tag{4.72}$$

where θ_{xy}^k is a diagonal matrix holding the two normalization factors given by

$$\theta_{xy}^k = \begin{pmatrix} \theta_x^k & 0 \\ 0 & \theta_y^k \end{pmatrix}. \tag{4.73}$$

Finally, to avoid a more complex minimization, we follow Demetz [55] and omit the coefficient increments $\mathbf{dc}^k$ within the gradient constancy assumption. Hence, the final differential formulation of the data term is given by

$$\begin{aligned} D(\mathbf{dw}^k, \mathbf{dc}^k) = \Psi_{\mathrm{c}}\Big(\theta^k \cdot \Big(\nabla I_2^{k\top}\mathbf{dw}^k + I_2^k - \Phi(I_1, \mathbf{c}^k + \mathbf{dc}^k)\Big)^2\Big) \\ + \gamma \cdot \Psi_{\mathrm{c}}\Big(\Big|\theta_{xy}^k\Big(\mathcal{H}(I_2^k)^\top \mathbf{dw}^k + \nabla I_2^k - \nabla\Phi(I_1, \mathbf{c}^k)\Big)\Big|^2\Big)\,. \end{aligned} \tag{4.74}$$

Regularization Terms In the case of the regularization terms, the differential formulation of the baseline is straightforward. The illumination coefficient regularizer reads

$$R_{\text{illum}}(\mathbf{dc}^k) = \sum_{l=1}^{2} \Psi_l \left(\sum_{i=1}^{N} \left(\mathbf{r}_l^\top \nabla (c_i^k + dc_i^k) \right)^2 \right), \tag{4.75}$$

and the original flow regularizer is given by

$$R_{\text{flow}}(\mathbf{dw}^k) = \Psi_c \left(\left| \mathcal{H}(u^k + du^k) \right|_{\text{F}}^2 + \left| \mathcal{H}(v^k + dv^k) \right|_{\text{F}}^2 \right). \tag{4.76}$$

The differential formulations of the other regularizers are derived analogously. Furthermore, in the case of the combined regularization and the indirect regularization, the splitting is also applied to the individual flow components and the auxiliary functions, respectively.

4.1.7.2 Minimality Conditions

After deriving the differential formulation the next step is to minimize the respective differential energies at each resolution level k. Therefore, we determine the minimality conditions, i.e., the Euler-Lagrange equations, corresponding to the particular prototypes. To keep things a little simpler we focus on the actual contributions from the different flow regularizers. To this end, we first specify the contributions of the data term and the illumination regularization term, which are the same for all the prototypes. Then we discuss the contributions of the flow regularizers.

Data Term Introducing the following two abbreviations for the non-linear expressions

$$\Psi'_{\text{data,bca}} := \Psi'_{\text{c}} \left(\theta^k \left(\nabla I_2^{k\top} \mathbf{dw}^k + I_2^k - \Phi(I_1, \mathbf{c}^k + \mathbf{dc}^k) \right)^2 \right), \tag{4.77}$$

$$\Psi'_{\text{data,gca}} := \Psi'_{\text{c}} \left(\left| \theta_{xy}^k \left(\mathcal{H}(I_2^k) \mathbf{dw}^k + \nabla I_2^k - \nabla \Phi(I_1, \mathbf{c}^k) \right) \right|^2 \right), \tag{4.78}$$

allows us to write the contributions related to the data term as follows

$$D_{\text{du}} := \Psi'_{\text{data,bca}} \theta^k \left(\nabla I_2^{k\top} \mathbf{dw}^k + I_2^k - \Phi(I_1, \mathbf{c}^k + \mathbf{dc}^k) \right) I_{2,x}^k \tag{4.79}$$

$$+ \Psi'_{\text{data,gca}} \left(\theta_{xy}^k \left(\mathcal{H}(I_2^k) \mathbf{dw}^k + \nabla I_2^k - \nabla \Phi(I_1, \mathbf{c}^k) \right) \right)^\top \nabla I_{2,x}^k , \tag{4.80}$$

$$D_{\text{dv}} := \Psi'_{\text{data,bca}} \theta^k \left(\nabla I_2^{k\top} \mathbf{dw}^k + I_2^k - \Phi(I_1, \mathbf{c}^k + \mathbf{dc}^k) \right) I_{2,y}^k \tag{4.81}$$

$$+ \Psi'_{\text{data,gca}} \left(\theta_{xy}^k \left(\mathcal{H}(I_2^k) \mathbf{dw}^k + \nabla I_2^k - \nabla \Phi(I_1, \mathbf{c}^k) \right) \right)^\top \nabla I_{2,y}^k , \tag{4.82}$$

$$D_{\text{dc1}} := \Psi'_{\text{data,bca}} \theta^k \left(\nabla I_2^{k\top} \mathbf{dw}^k + I_2^k - \Phi(I_1, \mathbf{c}^k + \mathbf{dc}^k) \right) (-\phi_1(I_1)) , \tag{4.83}$$

$$D_{\text{dc2}} := \Psi'_{\text{data,bca}} \theta^k \left(\nabla I_2^{k\top} \mathbf{dw}^k + I_2^k - \Phi(I_1, \mathbf{c}^k + \mathbf{dc}^k) \right) (-\phi_2(I_1)) . \tag{4.84}$$

Regularization Terms As for the data term contributions, we also introduce some useful abbreviations for the regularization terms. To this end, we adapt the diffusion tensor notation

from Subsection 4.1.6, such that it includes the iteration index k. For example, $T_{\text{1-iso}}$ shall denote the diffusion tensor for the isotropic first-order regularizer given by

$$T_{\text{1-iso}} := I_{2\times 2} \cdot \Psi' \left(\left| \nabla u^{k+1} \right|^2 + \left| \nabla v^{k+1} \right|^2 \right) \tag{4.85}$$

$$= I_{2\times 2} \cdot \Psi' \left(\left| \nabla (u^k + du^k) \right|^2 + \left| \nabla (v^k + dv^k) \right|^2 \right). \tag{4.86}$$

Analogously, we use the same notation for all the other diffusion tensors as well. To enhance the readability even further we also omit the iteration index k for the unknowns, e.g., $u := u^{k+1}$.

Euler-Lagrange Equations Next, we turn to the actual Euler-Lagrange equations. Depending on the applied regularization strategy the Euler-Lagrange equations form a system of four up to eight coupled non-linear partial differential equations. However, since all prototypes consider the same data term and the same regularizer for the illumination coefficients, the partial differential equations of the Euler-Lagrange equations related to the illumination coefficients are the same for all models. Hence, for the sake of clarity, we do not specify them repeatedly for all the different prototypes. Please keep in mind that the following two partial differential equations

$$0 = D_{\text{dc}_1} - \beta \cdot \nabla \cdot (T_{\text{1-illum}} \nabla c_1) \,, \tag{4.87}$$

$$0 = D_{\text{dc}_2} - \beta \cdot \nabla \cdot (T_{\text{1-illum}} \nabla c_2) \,, \tag{4.88}$$

with the boundary conditions

$$0 = \mathbf{n}^\top T_{\text{1-illum}} \nabla c_1 \,, \tag{4.89}$$

$$0 = \mathbf{n}^\top T_{\text{1-illum}} \nabla c_2 \,, \tag{4.90}$$

and the diffusion tensor

$$T_{\text{1-illum}} := \sum_{l=1}^{2} \Psi_l' \left(\sum_{i=1}^{2} \left(\mathbf{r}_l^\top \nabla (c_i^k + dc_i^k) \right)^2 \right) \cdot \mathbf{r}_l \, \mathbf{r}_l^\top \,, \tag{4.91}$$

are also part of each set of Euler-Lagrange equations that we discuss in the following.

First-Order Regularization In case of the first-order regularization strategies the corresponding Euler-Lagrange equations form a system of four coupled non-linear partial differential equations. The additional two equations for the flow components thereby read

$$0 = D_{\text{du}} - \alpha \cdot \nabla \cdot (T_1 \nabla u) \,, \tag{4.92}$$

$$0 = D_{\text{dv}} - \alpha \cdot \nabla \cdot (T_1 \nabla v) \,, \tag{4.93}$$

with the following boundary conditions

$$0 = \mathbf{n}^\top T_1 \nabla u \,, \tag{4.94}$$

$$0 = \mathbf{n}^\top T_1 \nabla v \,. \tag{4.95}$$

Depending if we consider the isotropic or anisotropic regularizer T_1 is either $T_{\text{1-iso}}$ or $T_{\text{1-aniso}}$.

Direct Second-Order Regularization For the direct second-order regularization, we have a system of four coupled non-linear partial differential equations. The additional two equations for the flow components are given by

$$0 = D_{\text{du}} + \alpha \cdot \nabla^2 \cdot \left(T_2 \, \nabla^2 u\right), \tag{4.96}$$
$$0 = D_{\text{dv}} + \alpha \cdot \nabla^2 \cdot \left(T_2 \, \nabla^2 v\right), \tag{4.97}$$

with the following associated boundary conditions

$$0 = \mathbf{n}^\top (\boldsymbol{\nabla}_2 \cdot T_2 \nabla^2 u), \tag{4.98}$$
$$0 = \mathbf{n}^\top (\boldsymbol{\nabla}_2 \cdot T_2 \nabla^2 v), \tag{4.99}$$
$$\mathbf{0} = \mathbf{n}_2^\top T_2 \nabla^2 u, \tag{4.100}$$
$$\mathbf{0} = \mathbf{n}_2^\top T_2 \nabla^2 v, \tag{4.101}$$

where $\mathbf{n}_2$ is defined as $\mathbf{n}_2 = (I_{2\times 2} \otimes \mathbf{n})$. Furthermore, as before, depending if we consider the isotropic or anisotropic variants T_2 is either $T_{\text{2-iso}}$, $T_{\text{2-aniso-s}}$ or $T_{\text{2-aniso-d}}$.

Combined Regularization In the case of the combined regularization, we implement the infimal convolution approach by splitting the unknowns not only in the regularization term but also in the data term, i.e., we replace $\mathbf{w}$ with $\mathbf{w} = \mathbf{w}_1 + \mathbf{w}_2$. While this does not change the actual minimizer, it simplifies the implementation. Hence, the additional Euler-Lagrange equations are given by

$$0 = D_{\text{du}} - \alpha \cdot \nabla \cdot (T_1 \, \nabla u_1), \tag{4.102}$$
$$0 = D_{\text{dv}} - \alpha \cdot \nabla \cdot (T_1 \, \nabla v_1), \tag{4.103}$$
$$0 = D_{\text{du}} + \alpha\lambda \cdot \nabla^2 \cdot \left(T_2 \, \nabla^2 u_2\right), \tag{4.104}$$
$$0 = D_{\text{dv}} + \alpha\lambda \cdot \nabla^2 \cdot \left(T_2 \, \nabla^2 v_2\right), \tag{4.105}$$

with the following boundary conditions

$$0 = \mathbf{n}^\top T_1 \nabla u_1, \tag{4.106}$$
$$0 = \mathbf{n}^\top T_1 \nabla v_1, \tag{4.107}$$
$$0 = \mathbf{n}^\top (\boldsymbol{\nabla}_2 \cdot T_2 \nabla^2 u), \tag{4.108}$$
$$0 = \mathbf{n}^\top (\boldsymbol{\nabla}_2 \cdot T_2 \nabla^2 v), \tag{4.109}$$
$$\mathbf{0} = \mathbf{n}_2^\top T_2 \nabla^2 u, \tag{4.110}$$
$$\mathbf{0} = \mathbf{n}_2^\top T_2 \nabla^2 v, \tag{4.111}$$

As before, depending if we consider the isotropic or anisotropic variants T_1 is either $T_{\text{1-inf-iso}}$ or $T_{\text{1-inf-aniso}}$ and T_2 is either $T_{\text{2-inf-iso}}$, $T_{\text{2-inf-aniso-s}}$ or $T_{\text{2-inf-aniso-d}}$.

Indirect Second-Order Regularization Finally, the indirect second-order regularization leads to a system of eight non-linear partial differential equations. The additional two equations for the flow components as well as the four additional equations for the auxiliary functions read

$$0 = D_{\text{du}} - \alpha \cdot \nabla \cdot (T_1^{\text{aux}}(\nabla u - \mathbf{a})) \,, \tag{4.112}$$
$$0 = D_{\text{dv}} - \alpha \cdot \nabla \cdot (T_1^{\text{aux}}(\nabla v - \mathbf{b})) \,, \tag{4.113}$$
$$\mathbf{0} = T_1^{\text{aux}}(\mathbf{a} - \nabla u) - \lambda \, \boldsymbol{\nabla}_2 \cdot (T_2^{\text{aux}} \boldsymbol{\nabla}_2 \mathbf{a}) \,, \tag{4.114}$$
$$\mathbf{0} = T_1^{\text{aux}}(\mathbf{b} - \nabla v) - \lambda \, \boldsymbol{\nabla}_2 \cdot (T_2^{\text{aux}} \boldsymbol{\nabla}_2 \mathbf{b}) \,, \tag{4.115}$$

with the associated boundary conditions given by

$$0 = \mathbf{n}^\top T_1^{\text{aux}}(\nabla u - \mathbf{a}) \,, \tag{4.116}$$
$$0 = \mathbf{n}^\top T_1^{\text{aux}}(\nabla v - \mathbf{b}) \,, \tag{4.117}$$
$$\mathbf{0} = \mathbf{n}_2^\top T_2^{\text{aux}}(\boldsymbol{\nabla}_2 \mathbf{a}) \,, \tag{4.118}$$
$$\mathbf{0} = \mathbf{n}_2^\top T_2^{\text{aux}}(\boldsymbol{\nabla}_2 \mathbf{b}) \,, \tag{4.119}$$

where the previous terms in Equation 4.114 and Equation 4.115 are in a vector-valued form to simplify the notation. Again, depending if we consider the isotropic or anisotropic variants T_1^{aux} is either $T_{\text{1-iso}}^{\text{aux}}$ or $T_{\text{1-aniso}}^{\text{aux}}$ and T_2^{aux} is either $T_{\text{2-iso}}^{\text{aux}}$, $T_{\text{2-aniso-s}}^{\text{aux}}$ or $T_{\text{2-aniso-d}}^{\text{aux}}$.

4.1.7.3 Numerical Solution

As in our minimization example of Subsection 2.4.2, the resulting Euler-Lagrange equations given in the previous section are non-linear in the unknowns. Hence, to resolve this difficulty, we approximate the individual non-linear system of equations through a sequence of linear systems of equations. In particular, we introduce a second fixed point iteration, where we keep the non-linear contributions lagging [36]. This approach leaves us with a linear system of equations in each fixed-point iteration. To solve this linear system of equations numerically, we discretize it on a regular grid. Furthermore, we apply the non-standard finite difference approximation of Weickert et al. [186] in the case of the first-order divergence expressions and the standard finite difference approximation in the case of the second-order divergence equivalents as well as the remaining derivatives. Finally, we solve the discretized system of equations using a cascadic [32] multicolor [16] variant of the SOR method [205].

4.1.8 Evaluation

After discussing the minimization process, we turn to the evaluation. As already mentioned we consider the KITTI 2012 [65] and KITTI 2015 [119] benchmark, since these benchmarks contain mainly scenes with highly non-fronto-parallel motion that constitute the main focus of interest.

Parameter Setting To aim for a fair comparison and to keep the number of adjustable parameters small, we set most parameters fixed and only optimized the weighting parameters α, β, γ, and λ for each prototype per benchmark and regularizer, see Subsection B.2.1. Furthermore, the penalizer functions of the flow regularizers were chosen to be the Charbonnier penalizer.

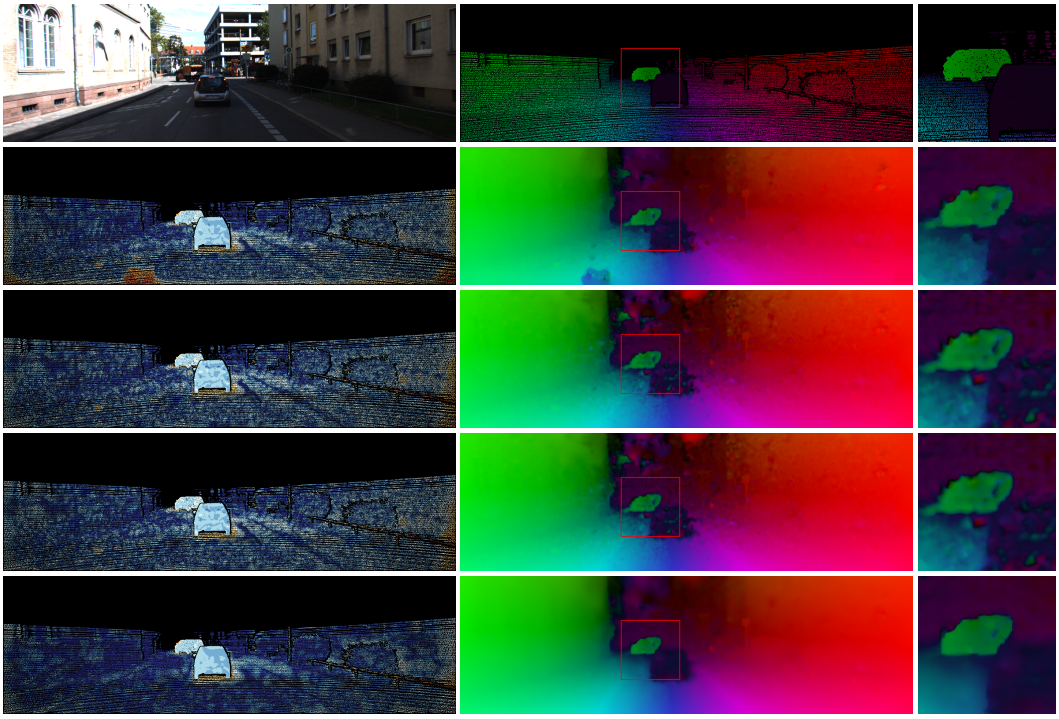


Figure 4.1: Sequence #166 of the KITTI 2015 benchmark [119]. *First row*: Reference frame, ground truth. *Following rows, from left to right*: Error and flow visualization. *Second row*: Anisotropic first-order regularizer. *Third row*: Double anisotropic direct second-order regularizer. *Fourth row*: Double anisotropic infimal convolution regularizer. *Fifth row*: Double anisotropic coupling regularizer.

First vs. Second-Order The first experiment points out the advantages of second-order regularization over the first-order regularization when it comes to the estimation of non-fronto-parallel motion. Therefore, we used an exemplary training sequence of the KITTI 2015 benchmark to compute the flow fields applying all the double anisotropic variants, including the first-order regularizer. The results in terms of both an error visualization and a flow visualization are depicted in Figure 4.1. Due to the special ego-motion of the camera, the pixels at the image boundary exhibit a large non-fronto-parallel motion which is not captured well with the first-order regularizer (second row). Nevertheless, it recovers quite sharp motion discontinuities.

In contrast, the smoothness weight α of the direct second-order (third row) as well as of the infimal convolution approach (fourth row) had to be chosen rather small to obtain a good result. This parameter choice, in turn, leads to noisy areas that are visible in the flow fields. The coupling model does not show this drawback (fifth row). It yields a result which is sharp at motion discontinuities, but smooth in homogeneous flow regions.

Furthermore, we included Figure 4.2, which shows the smoothing behavior of the first and the direct second-order regularization when increasing the smoothness, i.e., the weighting parameter α. While increasing smoothness leads to slightly blocky motion fields in the first-order case, this is not the case for the second-order regularization. However, for the direct second-order approach, the jumps are over-smoothed as well, which was not the case for the indirect second-order approach.

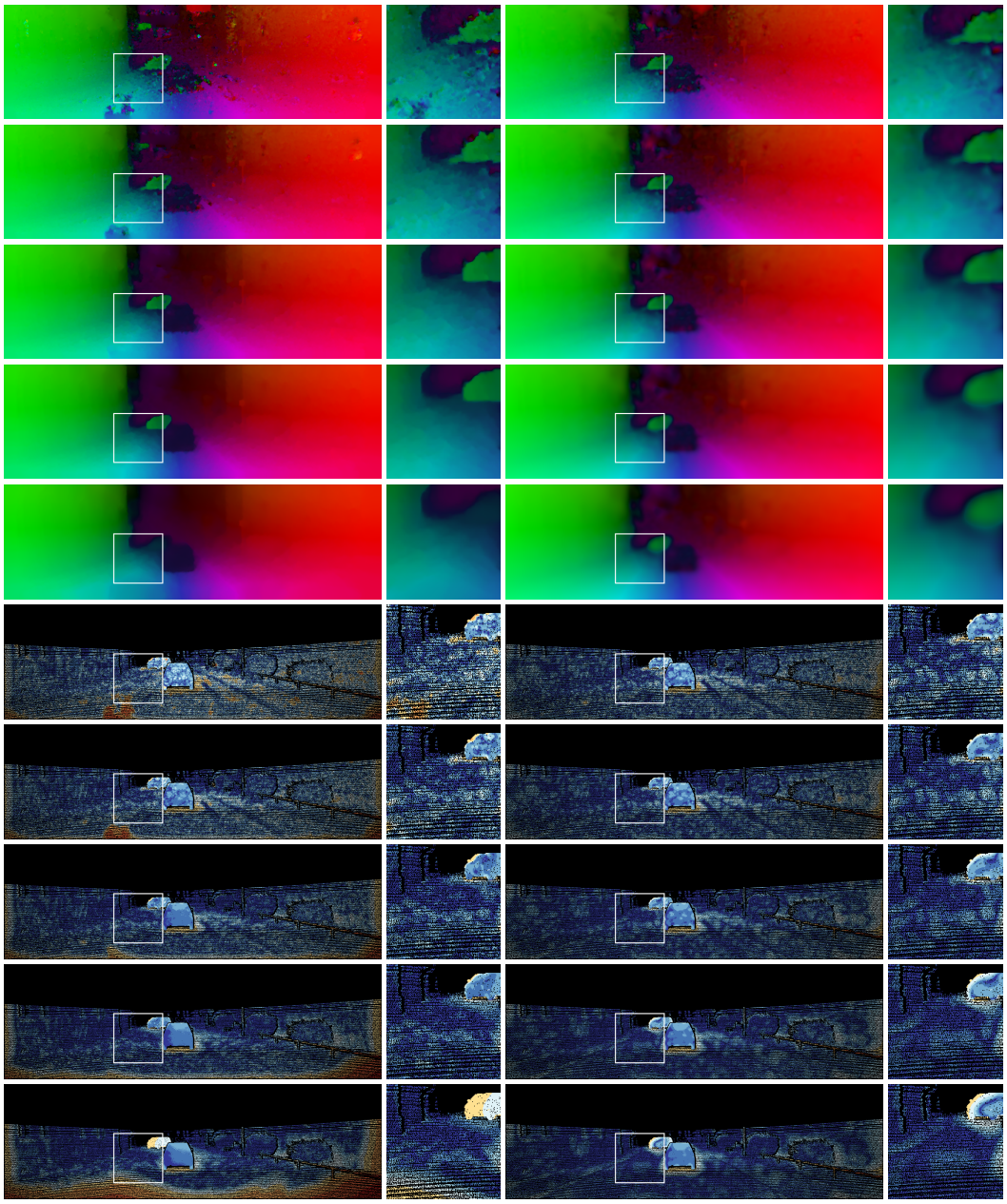


Figure 4.2: Training sequence #166 of the KITTI 2015 benchmark [119]. *Top to bottom*: Flow and error visualization with increased smoothness weight ($\alpha = 2, 4, 8, 16, 32$). *First column*: Anisotropic first-order regularizer. *Second column*: Double anisotropic direct second-order regularizer.

Table 4.1: Results for the KITTI 2012 [65] and KITTI 2015 [119] training data sets in terms of the average endpoint error (AEE), the percentage of erroneous pixels (BP) with a 3px threshold, and the runtimes for the different regularizers.

			KITTI 2012		KITTI 2015		
			AEE	BP	AEE	BP	runtime
direct	first-order	isotropic	4.39 px	16.53 %	12.88 px	29.87 %	**22** s
direct	first-order	anisotropic	4.26 px	16.25 %	12.45 px	28.94 %	23 s
direct	second-order	isotropic	2.75 px	10.07 %	11.40 px	24.61 %	26 s
direct	second-order	single anisotropic	2.35 px	9.84 %	10.60 px	24.29 %	33 s
direct	second-order	double anisotropic	2.32 px	9.73 %	10.49 px	24.05 %	33 s
inf-conv.	second-order	isotropic	2.30 px	9.66 %	9.60 px	23.61 %	43 s
inf-conv.	second-order	single anisotropic	2.30 px	9.64 %	9.45 px	23.30 %	51 s
inf-conv.	second-order	double anisotropic	2.28 px	9.58 %	9.38 px	23.00 %	52 s
coupling	second-order	isotropic	**2.18** px	9.65 %	9.24 px	22.90 %	65 s
coupling	second-order	single anisotropic	2.20 px	**9.57** %	9.07 px	22.37 %	65 s
coupling	second-order	double anisotropic	2.20 px	**9.57** %	**8.98** px	**22.27** %	96 s

Quantitative Evaluation In our second experiment, we not only used a single image sequence but the entire KITTI 2012 and 2015 training data sets to evaluate the performance of our regularizer prototypes. Therefore, we first optimized the respective model parameters w.r.t. the percentage of erroneous pixels (BP) using downhill simplex on a small subset of the training data and then evaluated the BP and the average endpoint error (AEE) on the entire training data set [14], see Table 4.1. In accordance with our first experiment, we see that second-order regularization is beneficial in the presence of non-fronto-parallel motion. One can also see, that introducing a higher degree of anisotropy allows reducing the errors further. In particular, for the direct second-order model this is obvious. We achieve the best performance in terms of accuracy with the double anisotropic coupling model. In terms of runtime, the increased complexity also leads to an increase in runtime. Especially, models with additional terms, i.e., infimal convolution models, models with additional unknowns, i.e. coupling models, and models with a higher degree of anisotropy, i.e., double anisotropic variants, require a longer computation time.

Different Penalizers Up to now, we have restricted our choice of the penalizer functions Ψ of the flow regularizers to the Charbonnier penalizer. Hence, in our third experiment, we consider the best performing prototype in terms of accuracy – the double anisotropic coupling model – and analyze different penalization strategies. To this end, we compare different combinations of the edge-enhancing Perona-Malik penalizer (PM) and the edge-preserving Charbonnier penalizer (Ch). The results in Table 4.2 show that choosing the leading penalizers Ψ_1 and $\Psi_{1,1}$ to be edge-enhancing works best. Only regarding the AEE of the KITTI 2015 benchmark, pure edge-preserving regularization allows achieving better results.

Comparison to the Literature Finally, in our fourth experiment, we compare our double anisotropic coupling model to other optical flow methods from the literature. To this end, we computed the flow fields of the KITTI 2012 and KITTI 2015 test data sets and submitted the

results to the online evaluation servers. In Table 4.3 and Table 4.4, we listed the best pure two-frame optical flow methods from the time of submission (8/9 Dec. 2016) that do not make use of additional information, such as stereo images, extra time-frames, semantic information or assume an underlying epipolar geometry. As one can see, our novel regularizer allows to obtain excellent results. In particular, it significantly outperforms all other solely variational approaches including BTF-ILLUM (our baseline model with the isotropic direct second-order [54]), TGV2ADCSIFT (isotropic second-order coupling [33]), and NLTGV-SC (non-local second-order coupling [137]). This observation, confirms once more the benefits of our double anisotropic coupling model.

Table 4.2: Comparison of different penalization strategies for the double anisotropic coupling model.

coupling term		smoothness term				KITTI 2012		KITTI 2015	
Ψ_1	Ψ_2	$\Psi_{1,1}$	$\Psi_{1,2}$	$\Psi_{2,1}$	$\Psi_{2,2}$	AEE	BP	AEE	BP
Ch	Ch	Ch	Ch	Ch	Ch	2.20 px	9.57 %	**8.98** px	22.27 %
PM	Ch	Ch	Ch	Ch	Ch	2.10 px	9.46 %	9.19 px	21.98 %
		PM	Ch	Ch	Ch	**2.05** px	**9.33** %	9.14 px	**21.82** %
		PM	PM	Ch	Ch	2.07 px	9.37 %	9.15 px	21.83 %
		PM	PM	PM	Ch	2.10 px	9.53 %	9.17 px	21.90 %

Table 4.3: Comparison of pure two-frame optical flow methods for the KITTI 2012 test sequences. Table shows the best performing methods at time of submission (8 Dec. 2016). Superscripts denote the rank of each method in the corresponding column.

Method	Out-Noc	Out-All	Avg-Noc	Avg-All
PatchBatch	5.29 % [1]	14.17 % [8]	1.3 px [1]	3.3 px [5]
our method	**5.57** % [2]	**10.71** % [2]	**1.3** px [1]	**2.8** px [1]
DDF	5.73 % [3]	14.18 % [9]	1.4 px [6]	3.4 px [6]
PH-Flow	5.76 % [4]	10.57 % [1]	1.3 px [1]	2.9 px [3]
FlowFields	5.77 % [5]	14.01 % [7]	1.4 px [6]	3.5 px [7]
CPM-Flow	5.79 % [6]	13.70 % [6]	1.3 px [1]	3.2 px [4]
NLTGV-SC	5.93 % [7]	11.96 % [4]	1.6 px [13]	3.8 px [9]
DDS-DF	6.03 % [8]	13.08 % [5]	1.6 px [13]	4.2 px [11]
TGV2ADCSIFT	6.20 % [9]	15.15 % [11]	1.5 px [9]	4.5 px [12]
DiscreteFlow	6.23 % [10]	16.63 % [12]	1.3 px [1]	3.6 px [8]
BTF-ILLUM	6.52 % [11]	11.03 % [3]	1.5 px [9]	2.8 px [1]
DeepFlow2	6.61 % [12]	17.35 % [14]	1.4 px [6]	5.3 px [13]
Data-Flow	7.11 % [13]	14.57 % [10]	1.9 px [15]	5.5 px [14]
DeepFlow	7.22 % [14]	17.79 % [15]	1.5 px [9]	5.8 px [15]
EpicFlow	7.88 % [15]	17.08 % [13]	1.5 px [9]	3.8 px [9]

Table 4.4: Comparison of pure two-frame optical flow methods for the KITTI 2015 test sequences. Table shows the best performing methods at time of submission (9 Dec. 2016). Superscripts denote the rank of each method in the corresponding column.

Method	FI-bg	FI-fg	FI-all
PatchBatch	19.98 %[1]	30.24 %[5]	21.69 %[1]
DDF	20.36 %[3]	29.69 %[3]	21.92 %[2]
our method	**20.01** %[2]	**32.82** %[6]	**22.14** %[3]
DiscreteFlow	21.53 %[4]	26.68 %[1]	22.38 %[4]
CPM-Flow	22.32 %[5]	27.79 %[2]	23.23 %[5]
Full-Flow	23.09 %[6]	30.11 %[4]	24.26 %[6]
EpicFlow	25.81 %[7]	33.56 %[7]	27.10 %[7]
DeepFlow	27.96 %[8]	35.28 %[8]	29.18 %[8]

4.1.9 Conclusion

In this section, we explored and compared several isotropic and anisotropic second-order regularization strategies for variational optical flow. In particular, we showed how different anisotropic variants can be derived from a single isotropic smoothness term, and how modeling a higher degree of anisotropy in terms of double anisotropic models can further improve the accuracy. Finally, experiments with the KITTI 2012 and 2015 benchmarks not only showed favorable results but also demonstrated that second-order coupling models, including the new double anisotropic regularizer, are among the state-of-the-art in the context of variational motion estimation.

4.2 An Order-Adaptive Regularization Strategy

So far we focused on second-order regularization techniques since they are known to be very beneficial when it comes to piecewise affine flow fields resulting from a moving camera. However, second-order priors also have a drawback. In particular, they are less suited to estimate fronto-parallel motion, since they are likely to misinterpret local fluctuations as affine motion (regularization order vs. robustness). This fact is also reflected in the most commonly used benchmarks. While leading variational methods in the automotive KITTI 2012 [65] and KITTI 2015 [119] benchmarks make use of second-order smoothness terms, the best performing variational approaches on the synthetic MPI Sintel [44] and Middlebury [23] benchmark rely on first-order priors.

It would be desirable to bridge this gap by developing an adaptive regularization strategy that selects the most appropriate regularization order. This adaptation would enable to combine the benefits of both techniques and hence allow to apply the corresponding approach to a broader range of applications and domains.

4.2.1 Related Work

The simplest way to implement such an adaptation is to resort to an user-based selection of the most appropriate regularization order [175]. This implementation, however, requires prior knowledge on the underlying application which might not always be available beforehand. Moreover, such

a hand-tuned strategy focuses on the entire application and hence does not allow to choose the most suitable order in a scene-wise or even pixel-wise manner.

Furthermore, one can consider second-order coupling models [33, 137] (indirect models), which we introduced in the previous section, as implicit adaptation techniques. Originally proposed in the context of denoising [34], the essential idea of such models is to add auxiliary functions that approximate first-order derivatives while imposing smoothness on these auxiliary functions themselves. Due to this design, coupling models allow the preservation of motion discontinuities in both the original and the auxiliary function. This behavior, in turn, comes down to edge-preserving first and second-order regularization, respectively. Although this strategy alleviates the problem of always performing second-order, one can observe that the performance of coupling models cannot keep up with the performance of first-order regularizers for benchmarks with mainly fronto-parallel motion (Middlebury, MPI Sintel). This fact clearly shows that there is room for improvement when it comes to the design of adaptive schemes for selecting the regularization order.

The only explicit order-adaptive approach that we found in the context of optical flow is the method of Volz et al. [179]. However, instead of selecting the spatial regularization order, the method determines the most appropriate order of the trajectorial regularization. Moreover, it relies on fitting polynomial approximations to the results of a preliminary estimation. In that sense, the selection process is not really self-contained, since it requires multiple estimations.

Finally, the work that we consider most similar in spirit to our method is the approach of Lenzen et al. [109]. Although the approach has been proposed in the context of denoising, the underlying variational model adaptively combines a direct first and second-order regularizer. Thereby the switching between the two regularizers is steered by local image information. While adapting the regularization order to the underlying image may be useful in the context of denoising, one should keep in mind that we are interested in motion estimation. Hence, slopes in the input images are typically unrelated to slopes in the flow field and hence do not provide useful information.

4.2.2 Contributions

In this section, we address all the shortcomings mentioned above: Based on an anisotropic first and an anisotropic second-order regularizer we derive a general strategy for variational methods, that allows adapting the regularization order automatically during the estimation. This strategy, that is based on the structural similarity of first-order models and second-order coupling models, not only allows to combine the advantages of both regularization orders, it is also solution-driven in that sense that the decision on the regularization order relies exclusively on the approximation quality of the resulting flow field. Based on this strategy, we propose four variational regularization techniques with a different degree of adaptation, ranging from a frame-wise adaptation to a pixel-wise selection. In this context, we also introduce concepts for spatially regularizing the decision process in terms of a non-local neighborhood or a global smoothness term. Finally, we demonstrate the usefulness of the proposed adaptation strategies by providing results for the most popular benchmarks. These results not only show the clear advantages of the global selection strategy compared to a standard single-order regularization, but it also makes explicit that a local selection strategy can be very beneficial if one combines it with some form of spatial coherence.

4.2.3 Baseline Model

As before, we seek to estimate the motion field $\mathbf{w} = (u, v)^\top : \Omega \to \mathbb{R}^2$ between two consecutive image frames $I_1, I_2 : \Omega \to \mathbb{R}$, where $\Omega \subset \mathbb{R}^2$ is the image domain, as the minimizer of an energy functional. The form of the considered energy functional is given by

$$E(\mathbf{w}) = D(\mathbf{w}) + \alpha \cdot R(\mathbf{w}) \,, \tag{4.120}$$

which consists of a data term D, a regularization term R, and a weighting parameter α.

Data Term To keep things simple and the focus on the flow regularization, we do not use the entire illumination-aware model of Demetz et al. [54] as before, but drop the illumination change estimation and compensation components, i.e., this choice comes down to the data term as proposed by Bruhn and Weickert [38]. Therefore, the data term combines a separately robustified brightness and gradient constancy assumption and is given by

$$D(\mathbf{w}) = \int_\Omega \Psi_c\Big(\big(I_2(\mathbf{x}+\mathbf{w}) - I_1(\mathbf{x})\big)^2\Big) + \gamma \cdot \Psi_c\Big(\big|\nabla I_2(\mathbf{x}+\mathbf{w}) - \nabla I_1(\mathbf{x})\big|^2\Big) d\mathbf{x} \,, \tag{4.121}$$

where $\mathbf{x} = (x, y)^\top \in \Omega$ denotes the location within the image domain Ω, γ is a weighting parameter to balance the two assumptions, and Ψ_c is the Charbonnier penalizer function.

Regularization Term In case of the regularization term R one typically employs either a first or a second-order regularization strategy. In the following, we will review a first-order regularizer as well as a second-order regularizer that we will combine in our order-adaptive framework later on. Our choice for these specific regularizers is motivated on the fact that they share essential structural properties and are thus particularly suited for a combination.

As *first-order regularizer*, we use the anisotropic complementary regularizer of Zimmer et al. [216]

$$R_1(\mathbf{w}) = \int_\Omega \underbrace{\sum_{l=1}^{2} \Psi_l\Big(\big(\mathbf{r}_l^\top \nabla u\big)^2 + \big(\mathbf{r}_l^\top \nabla v\big)^2\Big)}_{=S_1(\mathbf{w})} d\mathbf{x} \,. \tag{4.122}$$

As in the previous chapter $\mathbf{r}_1$ and $\mathbf{r}_2$ denote two spatially varying orthonormal vectors obtained as the eigenvectors of the regularization tensor [216], which can be considered a generalization of the structure tensor [59] to arbitrary constancy assumptions. Following Volz et al. [179], we apply the edge-enhancing Perona-Malik penalizer in case of Ψ_1 (across edges) and the edge-preserving Charbonnier penalizer in case of Ψ_2 (along edges).

Regarding the *second-order regularization* model we opt for the recent anisotropic coupling model of Hafner et al. [75], i.e., the single anisotropic coupling model as described in the previous section. It can be seen as an anisotropic variant of TGV [34] and is given by

$$R_2(\mathbf{w}) = \int_\Omega \inf_{\mathbf{a},\mathbf{b}} \Big\{ S_2(\mathbf{w}, \mathbf{a}, \mathbf{b}) + \lambda \cdot S_{\text{aux}}(\mathbf{a}, \mathbf{b}) \Big\} d\mathbf{x} \,. \tag{4.123}$$

It consists of two terms: the coupling term S_2 that connects the gradients ∇u and ∇v of the flow to the auxiliary functions $\mathbf{a} = (a_1, a_2)^\top$ and $\mathbf{b} = (b_1, b_2)^\top$ via

$$S_2(\mathbf{w}, \mathbf{a}, \mathbf{b}) = \sum_{l=1}^{2} \Psi_l \left(\left(\mathbf{r}_l^\top (\nabla u - \mathbf{a}) \right)^2 + \left(\mathbf{r}_l^\top (\nabla v - \mathbf{b}) \right)^2 \right), \tag{4.124}$$

and a smoothness term S_{aux} that enforces smoothness on these auxiliary functions themselves

$$S_{\text{aux}}(\mathbf{a}, \mathbf{b}) = \sum_{l=1}^{2} \Psi_l \left(\sum_{m=1}^{2} \left(\mathbf{r}_m^\top \mathcal{J}\mathbf{a}\, \mathbf{r}_l \right)^2 + \left(\mathbf{r}_m^\top \mathcal{J}\mathbf{b}\, \mathbf{r}_l \right)^2 \right). \tag{4.125}$$

Here $\mathcal{J}\mathbf{a}$ and $\mathcal{J}\mathbf{b}$ denote the Jacobian of $\mathbf{a}$ and $\mathbf{b}$, respectively, and the weighting parameter λ allows to adjust the smoothness of the auxiliary functions.

At this point, let us point out the substantial similarity between the first-order model S_1 and the coupling term S_2 of the second-order model. While the first-order model assumes the directional derivatives in $\mathbf{r}_1$ and $\mathbf{r}_2$-direction to be close to zero, the coupling term assumes them to be close to the auxiliary functions, which should be smooth by themselves. This similarity, in turn, makes the energies of both terms comparable and, consequently, makes them ideal candidates for a combination within our order-adaptive regularization framework.

4.2.4 Order-Adaptive Regularization

Having analyzed the structural similarity between the considered first and second-order regularizer, we are now in the position to introduce our order-adaptive regularization framework, that combines the advantages of both first and second-order regularization. In particular, we present four variants that differ in their degree of adaptivity, which ranges from a global (frame-wise) to a local (pixel-wise) adaptation.

Global Adaptive Scheme The first step to derive the order-adaptive framework is a simple combination of the previously introduced first and second-order regularizers, which reads

$$R(\mathbf{w}) = \int_\Omega \inf_{\mathbf{a},\mathbf{b}} \left\{ S_1(\mathbf{w}) + S_2(\mathbf{w}, \mathbf{a}, \mathbf{b}) + \lambda \cdot S_{\text{aux}}(\mathbf{a}, \mathbf{b}) \right\} d\mathbf{x}\,. \tag{4.126}$$

To implement the frame-wise adaption scheme, we introduce a weighting parameter $o \in (0, 1)$ and a selection term $\phi(o)$ similar in spirit to half-quadratic regularization [29]:

$$R(\mathbf{w}, o) = \int_\Omega \inf_{\mathbf{a},\mathbf{b}} \left\{ o \cdot S_1(\mathbf{w}) + (1-o) \cdot S_2(\mathbf{w}, \mathbf{a}, \mathbf{b}) + \lambda \cdot S_{\text{aux}}(\mathbf{a}, \mathbf{b}) + \phi_\theta(o) \right\} d\mathbf{x}\,. \tag{4.127}$$

Next, we derive a suitable selection term $\phi_\theta(o)$ that allows for a meaningful selection of the regularization order. In this context, it also becomes explicit why the convex combination only includes the first-order regularization term S_1 and the coupling term S_2 and not the smoothness term on the auxiliary functions S_{aux}. When deciding on the more suitable regularization order, the question naturally arises which model fits better: a constant model or an affine model. In the global adaptive scheme, we can answer this question by comparing the average energies related

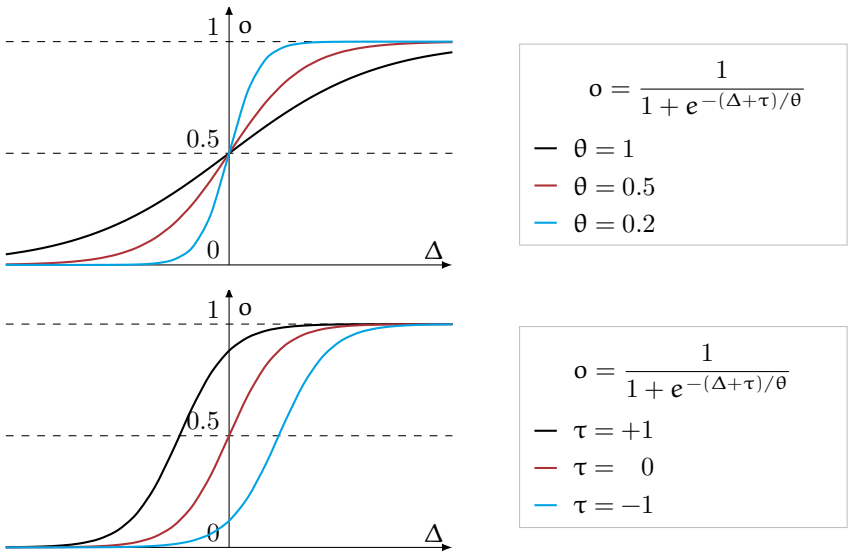


Figure 4.3: Plot of the sigmoid function for different θ values (top) and τ values (bottom).

to S_1 and S_2. Since the affine model S_2, however, includes the constant model S_1 one should only prefer the affine model S_2 if it yields a minimum average benefit τ compared to the constant model S_1. This minimum average benefit avoids overemphasizing small fluctuations, which would eliminate the advantage of the first-order regularizer, i.e., the robustness in case of nearly constant motion. Formulating this requirement in terms of a differentiable sigmoid function, we propose to determine the weighting parameter as

$$o = \frac{1}{1 + e^{-(\Delta+\tau)/\theta}} \quad \text{with} \quad \Delta = \frac{1}{|\Omega|} \int_\Omega S_2 - S_1 \, d\mathbf{x} \,, \tag{4.128}$$

where θ allows to adjust the slope, i.e., the sensitivity of the sigmoid function, see Figure 4.3 (top). As desired, o approaches a value of one if the average gain $\Delta \gg \tau$ and o approaches a value of zero if the average gain $\Delta \ll \tau$. Thereby, the minimum average benefit τ leads to a shift of the sigmoid function as shown in Figure 4.3 (bottom).

Let us now derive a selection term that models this desired behavior. To this end, we begin with the regularizer as given in Equation 4.127 and look at the derivative w.r.t. the weighting parameter $\partial_o R = 0$, which reads

$$0 = \int_\Omega S_1(\mathbf{w}) - S_2(\mathbf{w}, \mathbf{a}, \mathbf{b}) + \phi'_\theta(o) \, d\mathbf{x} \tag{4.129}$$

$$= \int_\Omega S_1(\mathbf{w}) - S_2(\mathbf{w}, \mathbf{a}, \mathbf{b}) \, d\mathbf{x} + \int_\Omega \phi'_\theta(o) \, d\mathbf{x} \tag{4.130}$$

$$= \int_\Omega S_1(\mathbf{w}) - S_2(\mathbf{w}, \mathbf{a}, \mathbf{b}) \, d\mathbf{x} + |\Omega| \cdot \phi'_\theta(o) \,. \tag{4.131}$$

By rearranging the differentiable sigmoid function given in Equation 4.128 as follows:

$$\frac{1}{1+e^{-(\Delta+\tau)/\theta}} = o \tag{4.132}$$

$$1+e^{-(\Delta+\tau)/\theta} = \frac{1}{o} \tag{4.133}$$

$$-(\Delta+\tau)/\theta = \ln\left(\frac{1}{o}-1\right) \tag{4.134}$$

$$-\Delta = \theta \cdot \ln\left(\frac{1}{o}-1\right)+\tau \tag{4.135}$$

$$-\frac{1}{|\Omega|}\int_{\Omega} S_2(\mathbf{w},\mathbf{a},\mathbf{b}) - S_1(\mathbf{w})\, d\mathbf{x} = \theta \cdot \ln\left(\frac{1}{o}-1\right)+\tau \tag{4.136}$$

$$\int_{\Omega} S_1(\mathbf{w}) - S_2(\mathbf{w},\mathbf{a},\mathbf{b})\, d\mathbf{x} = |\Omega| \cdot \left(\theta \cdot \ln\left(\frac{1}{o}-1\right)+\tau\right), \tag{4.137}$$

and plugging it into the derivative of the regularizer given in Equation 4.131 we obtain the following expression for the derivative of the selection term

$$\phi'_\theta(o) = -\theta \cdot \ln\left(\frac{1}{o}-1\right)-\tau\,. \tag{4.138}$$

Integrating both sides of this expression allows us to come up with a selection term that has the desired properties. Carrying out this integration yields

$$\phi_\theta(o) = \theta\left(\ln(1-o) - o \cdot \ln\left(\frac{1}{o}-1\right)\right) - \tau \cdot o + \text{constant}\,. \tag{4.139}$$

Now, we could simply plug in the selection function of Equation 4.139 into our model Equation 4.127. However, to obtain a more transparent write up of the model, we set the integration constant to be τ and introduce the following function

$$\phi(o) = \frac{\phi_\theta(o) - (1-o)\cdot\tau}{\theta} \tag{4.140}$$

$$= \ln(1-o) - o \cdot \ln\left(\frac{1}{o}-1\right) \tag{4.141}$$

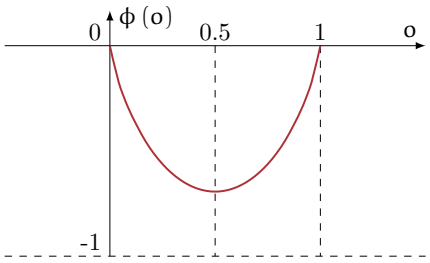
$$= \ln(1-o) - o \cdot \ln\left(\frac{1-o}{o}\right) \tag{4.142}$$

$$= \ln(1-o) - o \cdot \left(\ln(1-o) + \ln\left(\frac{1}{o}\right)\right) \tag{4.143}$$

$$= \ln(1-o) - o \cdot \ln(1-o) - o \cdot \ln\left(\frac{1}{o}\right) \tag{4.144}$$

$$= (1-o)\cdot\ln(1-o) + o \cdot \ln(o)\,, \tag{4.145}$$

which turns out to be the negative of the entropy function, as plotted in Figure 4.4. This particular



$$\phi(o) = (1-o)\cdot\ln(1-o) + o\cdot\ln(o)$$

Figure 4.4: Plot of the selection function $\phi(o)$.

selection function allows us to rewrite the global order-adaptive regularizer more intuitively as

$$R_{\text{global}}(\mathbf{w}, o) = \int_\Omega \inf_{\mathbf{a},\mathbf{b}} \Big\{ o\cdot S_1(\mathbf{w}) + (1-o)\cdot(S_2(\mathbf{w},\mathbf{a},\mathbf{b}) + \tau) + \lambda\cdot S_{\text{aux}}(\mathbf{a},\mathbf{b}) + \theta\cdot\phi(o) \Big\}\, d\mathbf{x}\,, \quad (4.146)$$

which can be regarded as a combination of a first-order regularizer and a second-order regularizer with activation cost τ, subject to a selection term with weight θ.

Local Adaptive Scheme Analogous to the global adaptive scheme, we can derive a local adaptive variant. The main difference is that we replace the global weighting parameter o with a spatially varying weighting function $o_{\text{local}} : \Omega \to (0,1)$. Furthermore, the new requirement on how to determine the weighting function relies on the local energy difference of S_1 and S_2 via

$$o_{\text{local}} = \frac{1}{1+e^{-(\Delta+\tau)/\theta}} \quad \text{with} \quad \Delta = S_2 - S_1\,. \quad (4.147)$$

The corresponding local adaptive regularizer reads

$$R_{\text{local}}(\mathbf{w}, o_{\text{local}}) = \int_\Omega \inf_{\mathbf{a},\mathbf{b}} \Big\{ o_{\text{local}}\cdot S_1(\mathbf{w}) + (1-o_{\text{local}})\cdot(S_2(\mathbf{w},\mathbf{a},\mathbf{b}) + \tau) + \lambda\cdot S_{\text{aux}}(\mathbf{a},\mathbf{b}) + \theta\cdot\phi(o_{\text{local}}) \Big\}\, d\mathbf{x}\,, \quad (4.148)$$

where we define the selection term ϕ as in case of the global scheme, see Equation 4.145. A similar weighting strategy has been used in the work of Xu et al. [199] to locally decide between two different constancy assumptions in the data term.

Non-Local Adaptive Scheme Besides the global approach, that performs a frame-wise adaption, and the local method, which operates on a location-wise basis, we further propose a variant that runs on an intermediate level, i.e., it takes a small neighborhood into account, when deciding on the regularization order. It is given by

$$R_{\text{non-local}}(\mathbf{w}, o_{\text{non-local}}) = \int_\Omega \inf_{\mathbf{a},\mathbf{b}} \Big\{ \bar{o}_{\text{non-local}}\cdot S_1(\mathbf{w}) + (1-\bar{o}_{\text{non-local}})\cdot(S_2(\mathbf{w},\mathbf{a},\mathbf{b}) + \tau) + \lambda\cdot S_{\text{aux}}(\mathbf{a},\mathbf{b}) + \theta\cdot\phi(o_{\text{non-local}}) \Big\}\, d\mathbf{x}\,, \quad (4.149)$$

where $\bar{o}_{\text{non-local}}$ integrates the actual weighting function $o_{\text{non-local}}$ over a small area via

$$\bar{o}_{\text{non-local}}(\mathbf{x}) = \frac{1}{|\mathcal{N}(\mathbf{x})|} \int_{\mathcal{N}(\mathbf{x})} o_{\text{non-local}}(\mathbf{y})\, d\mathbf{y}\,. \tag{4.150}$$

Here $\mathcal{N}(\mathbf{x})$ denotes a rectangular shaped neighborhood around $\mathbf{x}$ and $|\mathcal{N}(\mathbf{x})|$ is the size of the neighborhood, such that $\frac{1}{|\mathcal{N}(\mathbf{x})|} \int_{\mathcal{N}(\mathbf{x})} 1\, d\mathbf{y} = 1$. Furthermore, please note that we define $\mathcal{N}(\mathbf{x})$ in a way that only locations inside the image domain contribute to the area, i.e., $|\mathcal{N}(\mathbf{x})|$ becomes smaller towards image boundaries.

Employing the same selection term function ϕ as in the previous two cases, minimizing Equation 4.149 w.r.t. the weighting function $o_{\text{non-local}}$ yields

$$o_{\text{non-local}} = \frac{1}{1 + e^{-\Delta/\theta}} \quad \text{with} \quad \Delta = \int_{\mathcal{N}(\mathbf{x})} \frac{1}{|\mathcal{N}(\mathbf{y})|} (\tau + S_2 - S_1)\, d\mathbf{y}\,. \tag{4.151}$$

Furthermore, at locations where all neighborhoods have equal size, it further simplifies to

$$o_{\text{non-local}} = \frac{1}{1 + e^{-(\Delta+\tau)/\theta}} \quad \text{with} \quad \Delta = \frac{1}{|\mathcal{N}(\mathbf{x})|} \int_{\mathcal{N}(\mathbf{x})} (S_2 - S_1)\, d\mathbf{y}\,. \tag{4.152}$$

Therefore, the non-local approach can be seen as a generalization of our proposed adaptation scheme. In particular, it contains both the global and local variant, which constitute the largest and smallest choice of all possible neighborhoods.

Region Adaptive Scheme As a final variant, we propose a slightly different scheme, which also operates on an intermediate level. In contrast to the non-local approach, it does not integrate information over a neighborhood but employs smoothness constraints on the weighting function. Since it is not straightforward to realize this smoothness constraint in the same manner that we used so far, we pursue a level-set-based approach. In particular, we replace the selection cost term ϕ with a spatial smoothness term. Let $z : \Omega \to \mathbb{R}$ be a level-set function and let $o_{\text{region}}(z)$ denote a differential sigmoid function that approximates the Heaviside function

$$o_{\text{region}}(z) = \frac{1}{1 + e^{-z/\theta}}\,. \tag{4.153}$$

We propose the following region adaptive scheme

$$\begin{aligned} R_{\text{region}}(\mathbf{w}, z) = \int_\Omega \inf_{\mathbf{a},\mathbf{b}} \Big\{ & o_{\text{region}}(z) \cdot S_1(\mathbf{w}) + (1 - o_{\text{region}}(z)) \cdot (S_2(\mathbf{w}, \mathbf{a}, \mathbf{b}) + \tau) \\ & + \lambda \cdot S_{\text{aux}}(\mathbf{a}, \mathbf{b}) + \kappa \cdot |\nabla o_{\text{region}}(z)| \Big\}\, d\mathbf{x}\,. \end{aligned} \tag{4.154}$$

Let us point out that this scheme differs in several aspects from the other three variants. Firstly, the factor θ in front of the selection term determines the slope of the resulting sigmoid; in contrast, the weighting parameter κ in front of the smoothness term determines the amount of smoothing. Secondly, the region adaptive scheme comes with a slight drawback regarding the computational effort. While all previous approaches allow estimating the weighting parameter/function in closed

form, this is not possible for the region adaptive scheme. The reason is the spatial smoothness term, which requires to evolve the underlying level-set function z, see [18, 51].

4.2.5 Minimization

After proposing four different regularization models, we address their minimization. To this end, we employ the coarse-to-fine warping strategy, see Subsection 2.4.2.

4.2.5.1 Differential Formulation

The first step is to split the unknowns of each resolution level k of the coarse-to-fine scheme into the known intermediate solution and the unknown increment. However, in contrast to the previous implementations, we precede slightly different. While we perform the additive splitting for the flow field $\mathbf{w}$ as well as the auxiliary functions $\mathbf{a}$ and $\mathbf{b}$, we do not split the weighting parameter/function o, because we can explicitly compute its solution. Using the following splitting

$$\mathbf{w}^{k+1} = \mathbf{w}^k + \mathbf{dw}^k \,, \tag{4.155}$$

$$\mathbf{a}^{k+1} = \mathbf{a}^k + \mathbf{da}^k \,, \tag{4.156}$$

$$\mathbf{b}^{k+1} = \mathbf{b}^k + \mathbf{db}^k \,, \tag{4.157}$$

we can specify the differential energy related to the first three variants as follows

$$E(\mathbf{dw}^k, o^k) = D(\mathbf{dw}^k) + \alpha \cdot R(\mathbf{dw}^k, o^k) \,, \tag{4.158}$$

where D is the differential formulation of the data term, and R denotes a placeholder for the differential formulations of the regularizers corresponding to the different adaptation schemes. In the case of the region based strategy, which uses level-sets we also do not split the level-set function z. Hence, the differential formulation of the region based strategy is given by

$$E(\mathbf{dw}^k, z^k) = D(\mathbf{dw}^k) + \alpha \cdot R_{\text{region}}(\mathbf{dw}^k, z^k) \,, \tag{4.159}$$

where D is the same differential formulation of the data term as in Equation 4.158 and R_{region} is the differential formulation of the region based order-adaptive smoothness term.

Data Term As in the previous section, we obtain the differential formulation of the data term by linearizing the original expression w.r.t. the unknown flow increments. This linearization concerns $I_2(\mathbf{x} + \mathbf{w}^{k+1})$ in case of the brightness constancy assumption and $\nabla I_2(\mathbf{x} + \mathbf{w}^{k+1})$ in case of the gradient constancy assumption. Applying a first-order Taylor expansion results in

$$I_2(\mathbf{x} + \mathbf{w}^{k+1}) \approx \nabla I_2(\mathbf{x} + \mathbf{w}^k)^\top \mathbf{dw}^k + I_2(\mathbf{x} + \mathbf{w}^k) \,, \tag{4.160}$$

$$\nabla I_2(\mathbf{x} + \mathbf{w}^{k+1}) \approx \mathcal{H}(I_2(\mathbf{x} + \mathbf{w}^k))^\top \mathbf{dw}^k + \nabla I_2(\mathbf{x} + \mathbf{w}^k) \,. \tag{4.161}$$

Introducing the abbreviations $I_1 := I_1(\mathbf{x})$ and $I_2^k := I_2(\mathbf{x} + \mathbf{w}^k)$ allows writing the non-normalized differential formulation of the data term as

$$D(\mathbf{dw}^k) = \int_\Omega \Psi_\mathrm{c}\Big(\Big(\nabla I_2^{k\top}\mathbf{dw}^k + I_2^k - I_1\Big)^2\Big) + \gamma \cdot \Psi_\mathrm{c}\Big(\Big|\mathcal{H}(I_2^k)^\top \mathbf{dw}^k + \nabla I_2^k - \nabla I_1\Big|^2\Big)\, d\mathbf{x}\,. \quad (4.162)$$

Once more, we apply the constraint normalization [158, 216] to both constancy assumptions via the following normalization factors

$$\theta^k := \Big(|\nabla I_2^{k\top}|^2 + \zeta^2\Big)^{-\frac{1}{2}}, \quad (4.163)$$

$$\theta_x^k := \Big(|\nabla I_{2,x}^{k\top}|^2 + \zeta^2\Big)^{-\frac{1}{2}}, \quad (4.164)$$

$$\theta_y^k := \Big(|\nabla I_{2,y}^{k\top}|^2 + \zeta^2\Big)^{-\frac{1}{2}}, \quad (4.165)$$

where $\zeta = 0.01$ is a small parameter to prevent division by zero. Combining linearization and normalization, we obtain the normalized differential formulation as

$$D(\mathbf{dw}^k) = \Psi_\mathrm{c}\Big(\Big(\theta^k\Big(\nabla I_2^{k\top}\mathbf{dw}^k + I_2^k - I_1\Big)\Big)^2\Big) + \gamma \cdot \Psi_\mathrm{c}\Big(\Big|\theta_{xy}^k\Big(\mathcal{H}(I_2^k)^\top \mathbf{dw}^k + \nabla I_2^k - \nabla I_1\Big)\Big|^2\Big), \quad (4.166)$$

where θ_{xy}^k is a diagonal matrix that holds the two normalization factors given by

$$\theta_{xy}^k = \begin{pmatrix} \theta_x^k & 0 \\ 0 & \theta_y^k \end{pmatrix}. \quad (4.167)$$

Regularization Terms To keep the differential formulation of the smoothness term more compact, we first introduce the following abbreviations

$$S_1^k := \sum_{l=1}^{2} \Psi_l\Big(\Big(\mathbf{r}_l^\top \nabla u^{k+1}\Big)^2 + \Big(\mathbf{r}_l^\top \nabla v^{k+1}\Big)^2\Big), \quad (4.168)$$

$$S_2^k := \sum_{l=1}^{2} \Psi_l\Big(\Big(\mathbf{r}_l^\top\Big(\nabla u^{k+1} - \mathbf{a}^{k+1}\Big)\Big)^2 + \Big(\mathbf{r}_l^\top\Big(\nabla v^{k+1} - \mathbf{b}^{k+1}\Big)\Big)^2\Big), \quad (4.169)$$

$$S_\mathrm{aux}^k := \sum_{l=1}^{2} \Psi_l\Bigg(\sum_{m=1}^{2}\Big(\mathbf{r}_m^\top \mathcal{J}(\mathbf{a}^{k+1})\,\mathbf{r}_l\Big)^2 + \Big(\mathbf{r}_m^\top \mathcal{J}(\mathbf{b}^{k+1})\,\mathbf{r}_l\Big)^2\Bigg). \quad (4.170)$$

Please note that the superscript k of the abbreviation refers to the increments, e.g., du^k within $u^{k+1} = u^k + du^k$. Now, we can write the differential formulation of the four regularizers as

$$R_\mathrm{global}(\mathbf{dw}^k, o^k) = \int_\Omega \inf_{\mathbf{da}^k, \mathbf{db}^k} \Big\{ o^k \cdot S_1^k + (1 - o^k)\cdot(S_2^k + \tau) + \lambda \cdot S_\mathrm{aux}^k + \theta \cdot \phi(o^k)\Big\}\, d\mathbf{x}\,, \quad (4.171)$$

$$R_{\text{local}}(\mathbf{dw}^k, o^k_{\text{local}}) = \int_\Omega \inf_{\mathbf{da}^k;\mathbf{db}^k} \Big\{ o^k_{\text{local}} \cdot S^k_1 + (1 - o^k_{\text{local}}) \cdot (S^k_2 + \tau) + \lambda \cdot S^k_{\text{aux}} + \theta \cdot \phi(o^k_{\text{local}}) \Big\} \, d\mathbf{x}\,, \tag{4.172}$$

$$R_{\text{non-local}}(\mathbf{dw}^k, o^k_{\text{non-local}}) = \int_\Omega \inf_{\mathbf{da}^k;\mathbf{db}^k} \Big\{ \bar{o}^k_{\text{non-local}} \cdot S^k_1 + (1 - \bar{o}^k_{\text{non-local}}) \cdot (S^k_2 + \tau) + \lambda \cdot S^k_{\text{aux}} + \theta \cdot \phi(o^k_{\text{non-local}}) \Big\} \, d\mathbf{x}\,, \tag{4.173}$$

$$R_{\text{region}}(\mathbf{dw}^k, z^k) = \int_\Omega \inf_{\mathbf{da}^k;\mathbf{db}^k} \Big\{ o_{\text{region}}(z^k) \cdot S^k_1 + (1 - o_{\text{region}}(z^k)) \cdot (S^k_2 + \tau) + \lambda \cdot S^k_{\text{aux}} + \kappa \cdot |\nabla o_{\text{region}}(z^k)| \Big\} \, d\mathbf{x}\,. \tag{4.174}$$

4.2.5.2 Minimality Conditions

With the differential formulation at hand, we next derive the associated Euler-Lagrange equations for all four differential formulations. Therefore, we first specify the contributions of the data term, which are the same for all models, as well as the diffusion tensors related to the first and second-order regularizers. This not only keeps things clearly arranged but also allows us to focus on the contributions of the regularizers and the different adaptation schemes.

Data Term Contributions Introducing the following two abbreviations for the non-linear expressions within the differential energy of the data term (Equation 4.166)

$$\Psi'_{\text{data,bca}} := \Psi'_{\text{c}}\Big(\theta^k \Big(\nabla I_2^{k\top} \mathbf{dw}^k + I_2^k - I_1 \Big)^2 \Big)\,, \tag{4.175}$$

$$\Psi'_{\text{data,gca}} := \Psi'_{\text{c}}\Big(\Big| \theta^k_{xy} \Big(\mathcal{H}(I_2^k) \mathbf{dw}^k + \nabla I_2^k - \nabla I_1 \Big) \Big|^2 \Big)\,, \tag{4.176}$$

allows to write the contributions related to the data term as

$$D_{\text{du}} := \Psi'_{\text{data,bca}} \theta^k \Big(\nabla I_2^{k\top} \mathbf{dw}^k + I_2^k - I_1 \Big) I^k_{2,x} \tag{4.177}$$

$$+ \Psi'_{\text{data,gca}} \Big(\theta^k_{xy} \Big(\mathcal{H}(I_2^k) \mathbf{dw}^k + \nabla I_2^k - \nabla I_1 \Big) \Big)^\top \nabla I^k_{2,x}\,, \tag{4.178}$$

$$D_{\text{dv}} := \Psi'_{\text{data,bca}} \theta^k \Big(\nabla I_2^{k\top} \mathbf{dw}^k + I_2^k - I_1 \Big) I^k_{2,y} \tag{4.179}$$

$$+ \Psi'_{\text{data,gca}} \Big(\theta^k_{xy} \Big(\mathcal{H}(I_2^k) \mathbf{dw}^k + \nabla I_2^k - \nabla I_1 \Big) \Big)^\top \nabla I^k_{2,y}\,. \tag{4.180}$$

Diffusion Tensors To compactly state the contributions of the regularization terms and the different adaptation schemes, we make use of the diffusion tensor notation as introduced in Subsection 4.1.6. Therefore, we state the diffusion tensors associated to the anisotropic first-order regularizer and the indirect second-order regularizer, i.e., the diffusion tensors T_1, T_1^{aux}, and

T_2^{aux}, associated with the first-order smoothness term S_1, the coupling term S_2, and the auxiliary smoothness term S_{aux}, respectively. They are given by

$$T_1 = \sum_{l=1}^{2} \Psi_l' \left(\left(\mathbf{r}_l^\top \nabla u^{k+1} \right)^2 + \left(\mathbf{r}_l^\top \nabla v^{k+1} \right)^2 \right) \cdot \mathbf{r}_l \, \mathbf{r}_l^\top , \tag{4.181}$$

$$T_1^{\text{aux}} = \sum_{l=1}^{2} \Psi_l' \left(\left(\mathbf{r}_l^\top \nabla (u^{k+1} - \mathbf{a}^{k+1}) \right)^2 + \left(\mathbf{r}_l^\top (\nabla v^{k+1} - \mathbf{b}^{k+1}) \right)^2 \right) \cdot \mathbf{r}_l \, \mathbf{r}_l^\top , \tag{4.182}$$

$$T_2^{\text{aux}} = I_{2\times 2} \otimes \left(\sum_{l=1}^{2} \Psi_l' \left(\left| \mathcal{J} \mathbf{a}^{k+1} \, \mathbf{r}_l \right|^2 + \left| \mathcal{J} \mathbf{b}^{k+1} \, \mathbf{r}_l \right|^2 \right) \cdot \mathbf{r}_l \mathbf{r}_l^\top \right) . \tag{4.183}$$

Euler-Lagrange Equations After deriving the individual contributions, we can finally specify the Euler-Lagrange equations. For all order-adaptive variants, the first four non-linear partial differential equations of the system of equations are given by

$$0 = D_{\text{du}} - \alpha \cdot \left(\nabla \cdot \left(o \cdot T_1 \nabla u \right) + \nabla \cdot \left((1 - o) \cdot T_1^{\text{aux}} (\nabla u - \mathbf{a}) \right) \right) , \tag{4.184}$$

$$0 = D_{\text{dv}} - \alpha \cdot \left(\nabla \cdot \left(o \cdot T_1 \nabla v \right) + \nabla \cdot \left((1 - o) \cdot T_1^{\text{aux}} (\nabla v - \mathbf{b}) \right) \right) , \tag{4.185}$$

$$\mathbf{0} = o \cdot T_1^{\text{aux}} (\mathbf{a} - \nabla u) - \lambda \cdot \boldsymbol{\nabla}_2 \cdot \left(T_2^{\text{aux}} \boldsymbol{\nabla}_2 \mathbf{a} \right) , \tag{4.186}$$

$$\mathbf{0} = o \cdot T_1^{\text{aux}} (\mathbf{b} - \nabla v) - \lambda \cdot \boldsymbol{\nabla}_2 \cdot \left(T_2^{\text{aux}} \boldsymbol{\nabla}_2 \mathbf{b} \right) , \tag{4.187}$$

where we dropped the superscripts $k+1$ of u, v, $\mathbf{a}$ and $\mathbf{b}$ for the sake of clarity and depending if we consider the global, local, non-local or region based adaptation scheme o is either o, o_{local}, $o_{\text{non-local}}$ or o_{region}, respectively. Furthermore, depending on the different adaptation scheme, additional equations come along.

Global Adaptive Scheme In the case of the global adaption scheme, a single equation related to the weighting parameter o comes along, i.e.,

$$o = \frac{1}{1 + e^{-(\Delta + \tau)/\theta}} \quad \text{with} \quad \Delta = \frac{1}{|\Omega|} \int_\Omega S_2 - S_1 \, d\mathbf{x} . \tag{4.188}$$

Local Adaptive Scheme For the local scheme, we have multiple equations related to the weighting function o_{local} given by

$$o_{\text{local}} = \frac{1}{1 + e^{-(\Delta + \tau)/\theta}} \quad \text{with} \quad \Delta = S_2 - S_1 . \tag{4.189}$$

Non-Local Adaptive Scheme In the case of the non-local adaptation scheme, we have multiple equations related to the weighting function $o_{\text{non-local}}$ given by

$$o_{\text{non-local}} = \frac{1}{1 + e^{-(\Delta + \tau)/\theta}} \quad \text{with} \quad \Delta = \frac{1}{|\mathcal{N}(\mathbf{y})|} \int_{\mathcal{N}(\mathbf{x})} (S_2 - S_1) \, d\mathbf{y} . \tag{4.190}$$

Region Adaptive Scheme For the region based scheme, where we did not directly design the behavior, the additional equations that we have to solve for the level set function z is given by

$$0 = o'_{\text{region}}(z) \cdot \left(S_1 - (S_2 + \tau) - \kappa \, \nabla \cdot \left(\frac{\nabla z}{|\nabla z|} \right) \right). \tag{4.191}$$

4.2.5.3 Numerical Solution

With the Euler-Lagrange Equations at hand, we now address the numerical solution. Therefore, once again, to overcome the non-linearity we first introduce a second fixed-point iteration [36]. This second fixed-point iteration approximates the non-linear system of equations at each resolution level as a series of linear system of equations by keeping the non-linear expressions related to the data and smoothness term fixed, see Subsection 2.4.2. Next, to solve each of the linear systems of equations numerically, we first discretize them using a finite difference approximation. Finally, we apply a cascadic [32] multicolor [16] variant of the SOR method[205] to solve the discrete system of linear equations related to the increments of the flow field and the auxiliary functions.

Moreover, in case of the global, local and non-local scheme we update the weighting parameter o, the weighting functions o_{local} and $o_{\text{nl}}/\bar{o}_{\text{nl}}$ in the same fashion as the non-linear expressions. To this end, we explicitly evaluate the corresponding Equations (4.128), (4.147), and (4.150) and (4.151), respectively. Since we cannot compute the level-set function z in closed form, we employ an explicit scheme for its computation based on an upwind discretization [131].

4.2.6 Evaluation

Parameter Setting As in the last evaluation section we used a fixed set of parameters for our optimization scheme. The remaining parameters γ, α, λ, θ, τ and κ were set individually for each benchmark, see Subsection B.2.2.

Comparison of Selection Strategies In our first experiment, we investigate the performance of our introduced order-adaptive regularizers. Therefore, we created a set of synthetic image sequences, shown in Figure 4.5, that contain mainly fronto-parallel motion (sequence 4), affine motion (sequence 3), or a combination of both (sequence 1 and 2). Furthermore, to visually assess the order-adaption quality of the different regularizers, we made use of the gradient magnitude of the ground truth flow. It yields small values for fronto-parallel motion and large values for affine motion and thus may serve as a rough indicator of the present type of motion. The gradient magnitude, as well as the computed weighting maps o, o_{local}, $o_{\text{non-local}}$, o_{region} of the different selection schemes, are depicted in Figure 4.6. It shows the image-wise adaptation of the global approach as well as the pixel-wise adaptation of the other strategies. Moreover, one can see that the local approach exhibits a quite noisy selection behavior which is less prominent in the methods that consider additional neighborhood information or employ spatial smoothness constraints. Finally, one can observe that first-order regularization is preferred at motion discontinuities, while in other regions the order depends on the local fit.

Besides this qualitative evaluation, we also conducted a quantitative. Therefore, we chose a single parameter set per model by optimizing the average error of all sequences in terms of the average endpoint error (AEE). Table 4.5 shows the results. As one can see, regarding the average

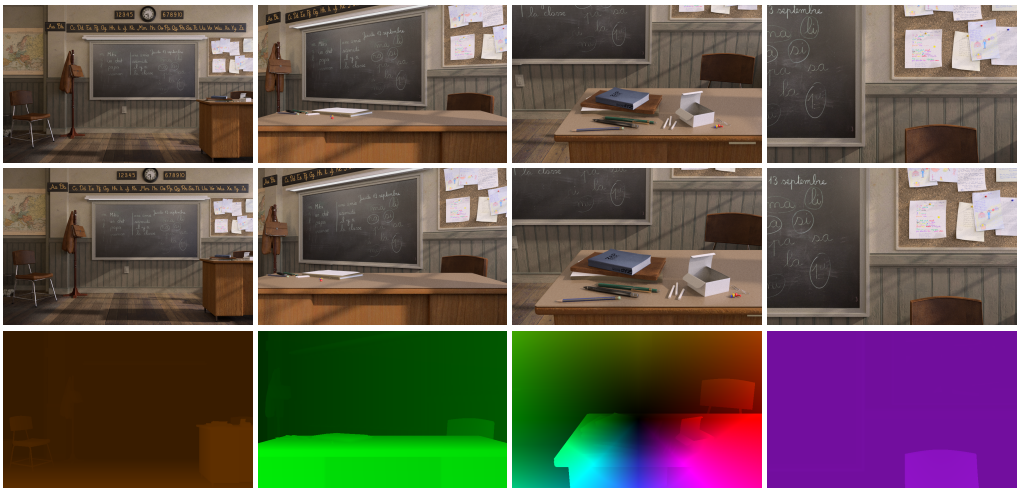


Figure 4.5: Synthetic Classroom sequences. *From left to right*: Sequence 1 to 4. *First row*: First frame. *Second row*: Second frame. *Third row*: Ground truth flow field.

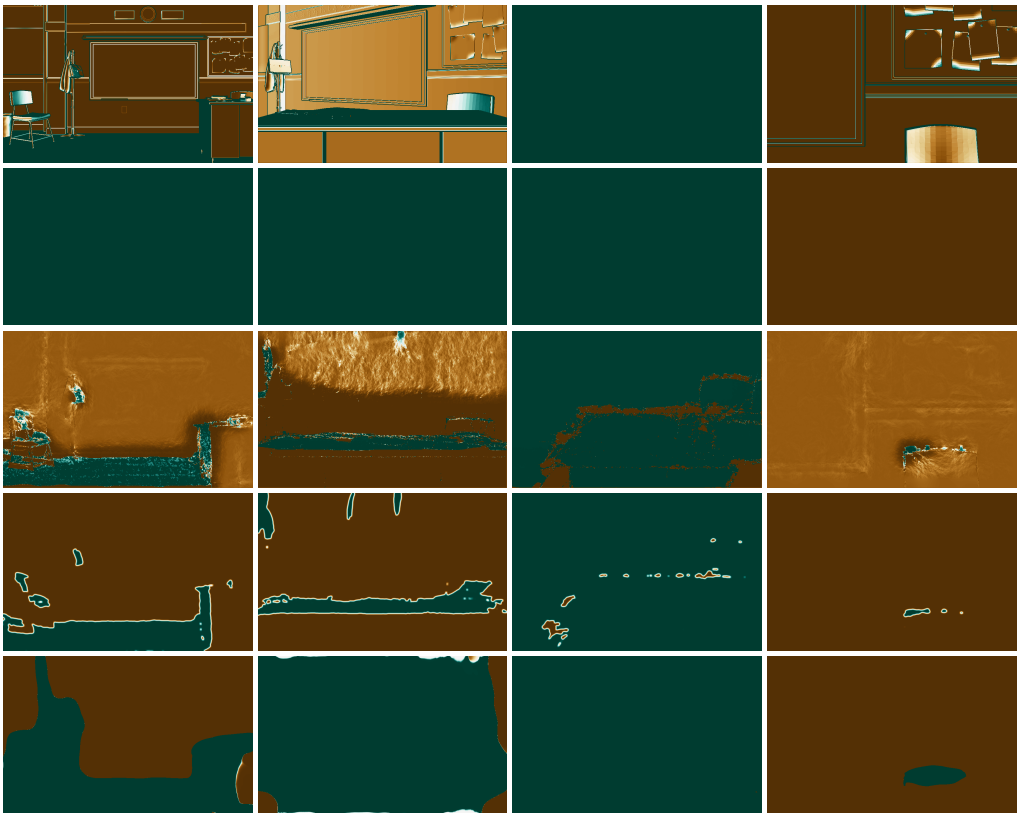


Figure 4.6: Adaptation behavior of the approaches in terms of the weighting map. *From left to right*: Sequence 1 to 4. *From top to bottom*: Gradient magnitude of the ground truth flow field, global approach o, local approach o_{local}, non-local approach $o_{\text{non-local}}$, and region-based approach o_{region}.

Table 4.5: Comparison of different regularization strategies for the four synthetic *Classroom* sequences in terms of the average endpoint error (AEE) and runtime.

		seq. #1	seq. #2	seq. #3	seq. #4	avg.	runtime
first-order		0.129 px	0.358 px	2.038 px	**0.088** px	0.653 px	**17** s
second-order		0.141 px	0.370 px	0.669 px	0.102 px	0.321 px	75 s
adaptive order	global	0.141 px	0.365 px	0.667 px	0.095 px	0.317 px	100 s
adaptive order	local	**0.111** px	**0.260** px	1.115 px	0.088 px	0.393 px	105 s
adaptive order	non-local	0.116 px	0.275 px	0.737 px	0.095 px	**0.307** px	120 s
adaptive order	region	0.125 px	0.366 px	**0.662** px	0.098 px	0.313 px	180 s

performance both non-adaptive approaches, i.e., the first-order regularizer and the second-order regularizer, are already outperformed by the global selection strategy. Please note that this strategy does not just come down to a frame-wise selection of the first and second-order results. On the one hand, it has to rely on the same parameters for both regularization orders, since the energy of both terms must be comparable to allow for a reasonable decision. On the other hand, the selection of the regularization order can be applied level-wise during the coarse-to-fine optimization, since this allows to correct less reliable decisions from coarse grid data. This behavior also explains why even for the individual error scores, the global decision strategy is sometimes able to outperform the results of the non-adaptive regularizers. Furthermore, recall that the considered second-order regularizer is an anisotropic variant of TGV [34]. This fact demonstrates that explicitly combining first and second-order smoothness terms in an adaptive manner allow us to outperform such implicit strategies based on robust coupling terms.

In contrast to the global strategy, the overall result of the local decision scheme does not seem very convincing. However, a closer look at the individual error values reveals that this strategy provides the best results for three out of four sequences. The main problem lies in the noisy selection behavior which results from the pixel-wise decision process. The non-local strategy and the region-based method provide the best overall performance. They combine the flexibility of the local approach with the robustness of the global strategy.

Comparison on Benchmark Data In our second experiment, we compare the four regularizers using the most popular optical flow benchmarks. To this end, we computed results for the training data sets of the Middlebury [23], the KITTI 2012 [65], the KITTI 2015 [119] and the MPI Sintel [44] benchmark. Following standard practice, we optimized the parameters per benchmark using the provided training data and the more common error metric, i.e., the AEE or the bad pixel measure (BP), respectively. Table 4.6 lists the corresponding results.

On the one hand, it becomes explicit that the order-adaptive strategies successfully combine the benefits of the simple first and second-order regularization methods. In most cases they even outperform the best non-adaptive result. On the other hand, the tendency of the different strategies confirms our observations from the first experiment. While the local approach can yield outstanding results (Sintel), it suffers from noisy decisions at the same time (KITTI 2012 and KITTI 2015). In contrast, the non-local scheme and the region-based scheme perform best. Also, the global se-

Table 4.6: Quantitative comparison of different regularization strategies for the four most popular benchmarks in terms of the average endpoint error (AEE) and the bad pixel error (BP).

		Middlebury AEE	Sintel AEE	KITTI 2012 BP	KITTI 2015 BP
first-order		0.213 px	4.327 px	18.026 %	30.053 %
second-order		0.222 px	6.518 px	9.461 %	22.736 %
adaptive order	global	0.211 px	4.213 px	9.423 %	22.424 %
adaptive order	local	0.211 px	**4.082** px	11.537 %	24.938 %
adaptive order	non-local	0.211 px	4.145 px	9.468 %	**22.158** %
adaptive order	region	**0.208** px	4.358 px	**9.415** %	22.343 %

lection strategy performs surprisingly well due to its robustness. Please note that by optimizing the activation cost τ jointly with the other parameters, the resulting decision scheme may favor a specific regularization order depending on the training data. However, the decision schemes still allow choosing between both regularization orders which significantly differs from just learning the regularization order. In this regard we also refer to the following chapter, i.e. Chapter 5, were we will see that one can choose a fixed activation cost τ across different data sets and still achieve excellent results.

Comparison to the Literature In our final experiment, we compare our results to related approaches from the literature. To this end, we submitted the results of our *non-local variant* to the public evaluation servers of all four benchmarks as mentioned earlier. The results in Table 4.7 show that we obtain similar results as comparable first-order methods, i.e., Zimmer et al. [216] on the Middlebury and the MPI Sintel benchmark. Furthermore, we achieve even better results than comparable second-order methods (i.e., Demetz et al. [54], Ranftl et al. [137]) on the KITTI 2012 and KITTI 2015 benchmark. This observation confirms that regularization order-adaptation is indeed worthwhile.

4.2.7 Conclusion

In this section, we investigated the usefulness of automatically adapting the regularization order in variational optical flow estimation. In this context, we proposed four different adaptation schemes together with four new order-adaptive regularizers that subtly fuse two anisotropic smoothness terms. Thereby, we introduced a local and a global selection strategy as well as a non-local and a region-based variant. While the global selection strategy turned out to be highly robust at the expense of being less adaptive, the local approach allowed a flexible point-wise selection at the cost of producing noisy decisions. By imposing some form of spatial regularity, i.e., neighborhood information or a spatial smoothness term, we finally succeeded to combine the advantages of both strategies. Our experiments confirmed these considerations. They showed that adaptively combining different regularization orders not only allows outperforming the non-adaptive strategy but also that in-frame-adaptivity may turn out useful if we regularize the decision process.

Table 4.7: Quantitative comparison of selected approaches for the four most popular benchmarks in terms of the average endpoint error (AEE) and the bad pixel error (BP).

benchmark	metric / sequence(s)	Our approach	Demetz et al. [54]	Ranftl et al. [137]	Zimmer et al. [216]
Middlebury	AEE / Army	**0.08** px	–	–	0.10 px
	/ Mequon	0.26 px	–	–	**0.19** px
	/ Schefflera	**0.38** px	–	–	0.43 px
	/ Wooden	**0.16** px	–	–	0.17 px
	/ Grove	**0.83** px	–	–	0.87 px
	/ Urban	**0.31** px	–	–	0.43 px
	/ Yosemite	**0.08** px	–	–	0.10 px
	/ Teddy	**0.52** px	–	–	0.59 px
KITTI 2012	BP / non-occ.	**5.69** %	6.52 %	5.93 %	–
	/ all	**10.72** %	11.03 %	11.96 %	–
	AEE / non-occ.	**1.4** px	1.5 px	1.6 px	–
	/ all	**2.8** px	**2.8** px	3.8 px	–
KITTI 2015	BP / bg	**20.62** %	–	–	–
	/ fg	**27.67** %	–	–	–
	/ all	**21.79** %	–	–	–
Sintel (final)	AEE / all	**8.179** px	–	8.746 px	8.204 px
	/ non-occ.	4.578 px	–	4.635 px	**4.448** px
	/ occluded	**37.525** px	–	42.242 px	38.805 px
Sintel (clean)	AEE / all	**6.227** px	–	7.680 px	6.496 px
	/ non-occ.	**2.760** px	–	3.565 px	2.849 px
	/ occluded	**34.455** px	–	41.168 px	36.216 px

4.3 Summary

In this chapter, we first identified suitable second-order regularization approaches for variational optical flow by comparing several different modeling strategies. Subsequently, we combined two regularizers of first and second-order in a sophisticated way to improve upon existing regularization techniques and proposed our new order-adaptive regularizer. This new regularizer shows a great flexibility and removes the need to decide on a specific regularization order prior to the estimation. Consequently, it offers an ideal choice for variational optical flow estimation.

5 Beyond Variational Motion Estimation

5.1 Introduction

So far we focused on purely variational methods to solve the motion estimation problem. To render the optimization feasible, one typically applies a linearization of the highly non-convex data term. This linearization, however, complicates the estimation of large displacements, since it is generally only valid for small displacements. To cope with this issue, we made use of the warping strategy [36], as described in Subsection 2.4.2. While this improves the estimation of large displacements, it does not resolve the problem for small objects that undergo a large displacement, since they disappear on coarser resolution levels. To deal with these large displacements of small objects, researchers proposed different solutions, e.g., the integration of point correspondences obtained via a preceding descriptor matching step [37, 159, 191] or the embedding of additional candidate matches to improve the initialization at each coarse-to-fine level [168, 199]. Another approach is to replace the coarse-to-fine scheme by a proper initialization, obtained via a sparse-to-dense interpolation of point correspondences [143]. In fact, most state-of-the-art large displacement optical flow pipelines use the latter method and refer to the variational component as variational refinement [20, 50, 61, 85, 120].

Since the variational refinement plays an essential role in many recent motion estimation approaches, it is surprising that most of those methods rely on rather simple models for the refinement. In particular, the employed refinement strategy typically cannot keep up with the adaptivity and robustness of the preceding pipeline steps – descriptor matching, filtering, and inpainting – which are typically rather elaborated. The most prominent example is the widely used refinement model of the EpicFlow pipeline [143] that essentially combines a classical gradient constancy assumption with a simple isotropic first-order smoothness term. In the last few years, however, there has been significant progress in the modeling of variational methods. This progress includes more advanced data terms with a higher degree of invariance [53, 112, 121, 140], the joint estimation of motion and illumination changes [54], higher-order regularizers [33, 9, 137], as well as anisotropic [9, 216] and non-local smoothness terms [137, 192]. All those developments address significant real-world problems such as robustness under varying illumination, the estimation of motion induced by a moving camera, or the sharp separation of motion boundaries – problems that are also reflected in recent motion estimation benchmarks [44, 65, 119]. Hence, it is quite surprising that there have been no attempts in the literature so far to develop variational methods for optical flow refinement that consider these advanced concepts.

In this chapter, we introduce such an advanced variational refinement scheme based on recent concepts. In particular, one of the key ingredients is our order-adaptive regularization that we

introduced in the previous chapter, see Section 4.2. This choice allows us to come up with a refinement method that eliminates one of the major shortcomings of current refinement models. Main parts of this chapter are based on the work published in [8].

5.2 Related Work

Conceptually closest related to our overall approach is the EpicFlow pipeline by Weinzaepfel et al. [191] as well as several follow-up works based on this pipeline. While most of these works focus on improving the matching step [20, 50, 61, 85, 120], there have hardly been any attempts to improve the sparse-to-dense inpainting [84, 219] and, to the best of our knowledge, no efforts to improve the refinement step. Apart from that, also recent works based on discrete optimization also make use of variational refinement [50, 120]. While these approaches do not necessarily suffer from the large displacement problem, they typically do not provide sub-pixel precise flow fields. To alleviate the latter shortcoming, they also resort to a variational refinement in terms of post-processing. Finally, from a variational viewpoint, closest related to our work are the works of Demetz et al. [54] that already served as a baseline method in the previous chapter, see Subsection 4.1.2, and our work on order-adaptive regularization [9] presented in the Section 4.2, which both use a traditional coarse-to-fine minimization scheme. These methods, however, have difficulties in dealing with fine structures and large displacements.

5.3 Contributions

In this chapter, we propose a novel model for variational refinement that combines robustness under varying illumination with our novel adaptive estimation of higher-order motion fields. To this end, we use an illumination-aware data term that is able to cope with locally affine illumination changes together with our anisotropic order-adaptive regularizer form the previous chapter that is able to produce solutions with gradual transitions where necessary while preserving sharp motion discontinuities at the same time. Moreover, we suggest a reduced coarse-to-fine scheme that is able to benefit from a good initialization within the pipeline approach while still being able to correct errors in the intermediate results. The benefits of our new refinement method become explicit in the experimental evaluation. The experiments not only show improvements compared to conventional refinement schemes and pure variational methods, they also demonstrate good results on all major benchmarks such as KITTI 2012 [65], KITTI 2015 [119] and MPI Sintel [44].

5.4 Pipeline for Large Displacement Optical Flow

Many state-of-the-art methods for large displacement optical flow use a pipeline approach as presented in [143], which is composed of four main steps: matching, outlier filtering, inpainting, and variational refinement. Figure 5.1 illustrates these four steps of the pipeline. In the following, we will detail each of these four steps as they also form the basis of our algorithm.

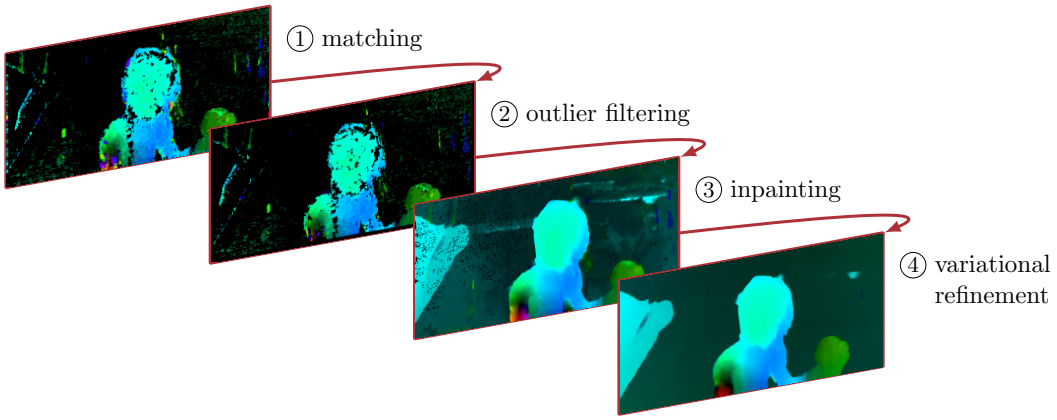


Figure 5.1: Illustration of the commonly used pipeline for large displacement optical flow by the example of a sequence from the MPI Sintel benchmark [44].

5.4.1 Matching

The goal of the first step in the pipeline is the generation of input matches. Generally, one can use all kinds of different algorithms, but as a matter of course, the matches should be rather dense, so that one obtains a reasonable initialization.

In our work we consider three different approaches to obtain input matches, of which all are tailored to the problem of optical flow estimation. Our first choice is the Deep Matching approach [191], which creates matches by computing similarities of non-rigid patches. It is based on a hierarchical, multi-layer, correlational architecture (inspired by deep convolutional approaches but not learning based) and is the favored choice in the work [143]. Our second choice is the recent CPM method [85] – a coarse-to-fine variant of PatchMatch [25] – which comes down to an approximate nearest neighbor field algorithm with an implicit regularization. To measure the similarities of matches the CPM algorithm makes use of SIFT features [114]. Our last choice is DiscreteFlow [120]. In contrast to the other two approaches, it contains explicit regularization. To obtain the matches, it first extracts a set of suitable proposals and optimizes a cost function via dynamic programming. Like the other approaches, it makes use of a robust feature descriptor, in this case, DAISY [166].

5.4.2 Outlier Filtering

The computed matches from the first step typically contain a certain amount of outliers, which occur for example due to occluded or low textured image regions. Since such erroneous matches can deteriorate the estimation substantially, it is essential to perform some sort of outlier filtering such as bidirectional consistency checking and removal of small isolated segments. In practice, this second step does not eliminate all outliers, but considerably reduces their amount. In our approach, we stick to the filtering steps as proposed by the respective matching approaches [85, 120, 191].

5.4.3 Inpainting

After removing outliers, the resulting flow field is typically non-dense. However, since the last step of the pipeline – the variational refinement – requires a dense flow field for initialization, the missing locations have to be inpainted. For this purpose, we use the locally-weighted affine variant of the Edge-Preserving Interpolation of Correspondences (EpicInpainting) as presented in [143], which locally fits an affine transformation to estimate missing flow vectors. The algorithm applies a weighted least-squares fit, where the weights are determined using a geodesic distance based on the image edges which are assumed to be a superset of the motion boundaries.

5.4.4 Variational Refinement

The final step refines the inpainted flow field using a variational method. Typically, this step aims at obtaining sub-pixel precision while it additionally introduces some regularization. In our case, we investigate and compare two variational models that we explain in the following sections: the commonly used EpicFlow [143] model that serves as a baseline in our evaluation and our novel order-adaptive illumination-aware model.

5.5 The EpicFlow Refinement Model

Let us start by discussing the commonly used EpicFlow refinement model [143]. To this end, let $I_1, I_2 : \Omega \to \mathbb{R}$ denote two consecutive frames of an image sequence defined on the rectangular image domain $\Omega \subset \mathbb{R}^2$. Furthermore, let $\mathbf{w} = (u, v)^\top : \Omega \to \mathbb{R}^2$ be the motion field we aim to estimate. Then the EpicFlow model computes the refined flow as a minimizer of an energy functional of the form

$$E_{\text{epic}}(\mathbf{w}) = \int_\Omega D_{\text{epic}}(\mathbf{w}) + \alpha \cdot R_{\text{epic}}(\mathbf{w}) \, d\mathbf{x}\,, \tag{5.1}$$

where D_{epic} and R_{epic} denote the data term and the regularization term, respectively, and α is the weighting parameter that steers the relative impact of both terms.

Data Term The data term D_{epic} comprises a brightness constancy assumption and a gradient constancy assumption with an additional constraint normalization as proposed in [216]. The data term equates to the model that we used in Section 4.2 with included constraint normalization, which reads

$$D_{\text{epic}}(\mathbf{w}) = \Psi_r\Big(\big(\theta(I_2(\mathbf{x}+\mathbf{w}) - I_1(\mathbf{x}))\big)^2\Big) + \gamma \cdot \Psi_r\Big(\big|\theta_{xy}(\nabla I_2(\mathbf{x}+\mathbf{w}) - \nabla I_1(\mathbf{x}))\big|^2\Big)\,, \tag{5.2}$$

Here $\mathbf{x} = (x, y)^\top$ denotes a position within the rectangular image domain $\Omega \subset \mathbb{R}^2$, Ψ_r the regularized linear penalizer function, and θ, θ_{xy} are normalization factors defined as

$$\theta = \frac{1}{\sqrt{|\nabla I_2^\top|^2 + \zeta^2}}\,, \qquad \theta_{xy} = \begin{pmatrix} \frac{1}{\sqrt{|\nabla I_{2,x}^\top|^2+\zeta^2}} & 0 \\ 0 & \frac{1}{\sqrt{|\nabla I_{2,y}^\top|^2+\zeta^2}} \end{pmatrix}, \tag{5.3}$$

where the parameter ζ not only avoids a division by zero but also reduces the influence of small gradients, e.g., noise in flat regions. While this data term works quite well in practice, it comes with the slight drawback that the gradient constancy assumption only allows handling additive illumination changes but not multiplicative ones – in contrast to the descriptors of the initial matching process; see e.g., SIFT [114] or DAISY [166] descriptors.

Regularization Term In case of the regularization term, R_{epic} the EpicFlow model uses an isotropic first-order flow-driven model with an image based weighting similar to [182], given by

$$R_{\text{epic}}(\mathbf{w}) = g(|\nabla I_1)|) \cdot \Psi_r\left(|\nabla u|^2 + |\nabla v|^2\right) \quad \text{with} \quad g(|\nabla I_1|) = \exp(-\kappa \cdot |\nabla I_1|)\,. \tag{5.4}$$

Here the robust penalizer function Ψ_r allows to preserve motion discontinuities and the spatially adaptive weight g tries to align these discontinuities with image boundaries, i.e., it reduces the impact of the smoothness term at image edges depending on the parameter κ. This smoothness term has two major drawbacks: On the one hand, since it uses the first-order regularization that prefers piecewise constant flow fields, it has problems with estimating highly non-fronto-parallel motion, e.g., an affine motion that is typically present in ego-motion scenes. On the other hand, it does not make use of directional information to refine motion boundaries, which usually gives a less distinct separation of objects in the flow field compared to smoothness terms based on anisotropic regularization.

5.6 Order-Adaptive Illumination-Aware Refinement Model

Based on the drawbacks of the EpicFlow model, we propose a novel order-adaptive and illumination-aware refinement model that combines recent concepts of variational optical flow estimation to eliminate these shortcomings. The two main concepts for this purpose we, actually, already introduced in the previous chapter: the illumination-aware data term of Demetz et al. [54] explained in Subsection 4.1.2 and our novel order-adaptive regularizer presented in Section 4.2. Hence, we revise the concepts briefly but clearly. To this end, we start with the data term proposed of Demetz et al. [54], that explicitly models local illumination changes in terms of a set of coefficient fields $\mathbf{c} = (c_1, \dots, c_n)^\top : \Omega \to \mathbb{R}^n$, and then turn to our recent anisotropic order-adaptive regularizer, that locally selects between first and second-order regularization using a spatially varying weighting function $o : \Omega \to (0, 1)$. Our new refinement model has the following form

$$E_{\text{oir}}(\mathbf{w}, \mathbf{c}, o) = \int_\Omega D_{\text{illum}}(\mathbf{w}, \mathbf{c}) + \alpha \cdot R_{\text{oar}}(\mathbf{w}, o) + \beta \cdot R_{\text{illum}}(\mathbf{c})\, d\mathbf{x}\,, \tag{5.5}$$

where D_{illum} is the illumination-aware data term, R_{oar} is the order-adaptive regularizer, and R_{illum} is the coefficient regularizer with the two weighting parameters α and β. Let us now detail the different components of the energy, starting with the data term.

Data Term As in the EpicFlow model, the data term consists of a brightness and gradient constancy assumption. To account for more general illumination changes, however, it additionally

uses a parametrized brightness transfer function $\Phi(I, \mathbf{c})$ in both assumptions [71]. The resulting illumination-aware data term reads

$$D_{\text{illum}}(\mathbf{w}, \mathbf{c}) = \Psi_{\text{c}}\Big(\big(\theta(I_2(\mathbf{x}+\mathbf{w}) - \Phi(I_1(\mathbf{x}), \mathbf{c}))\big)^2\Big) + \gamma \cdot \Psi_{\text{c}}\Big(\big|\theta_{xy}(\nabla I_2(\mathbf{x}+\mathbf{w}) - \nabla\Phi(I_1(\mathbf{x}), \mathbf{c}))\big|^2\Big)\,, \tag{5.6}$$

where Ψ_c is the Charbonnier penalizer and θ, θ_{xy} are again the normalization factors defined as before, see Equation 5.3. The general parametrized brightness transfer function [54, 71] is given by

$$\Phi(I, \mathbf{c}) = \bar{\phi}(I) + \sum_{i=1}^{n} c_i \cdot \phi_i(I)\,, \tag{5.7}$$

where $\phi_i(I) : \mathbb{R} \to \mathbb{R}$ denote the n basis functions and $\bar{\phi}(I) : \mathbb{R} \to \mathbb{R}$ is the mean brightness transfer function. As in the previous chapter, see Subsection 4.1.2, we choose $\Phi(I, \mathbf{c})$ to be the normalized affine function, i.e.,

$$\bar{\phi}(I) = I\,, \quad \phi_1(I) = \frac{I}{n_1}\,, \quad \text{and} \quad \phi_2(I) = \frac{1}{n_2}\,, \tag{5.8}$$

where n_1 and n_2 are normalization factors such that $\|\phi_i(I)\|_2 = 1$. Compared to a learned brightness transfer function, this choice not only offers an intuitive interpretation of the coefficient fields but also is suitable for a broad variety of different domains.

Regularization Term (Illumination) In the case of the regularizer for the illumination coefficients we follow [54] and use a joint anisotropic first-order regularizer which reads

$$R_{\text{illum}}(\mathbf{c}) = \sum_{l=1}^{2} \Psi_l\left(\sum_{i=1}^{n}\left(\mathbf{r}_l^\top \nabla c_i\right)^2\right). \tag{5.9}$$

It not only enables the regularization to locally adapt to the underlying image structure in terms of two spatially varying directions $\mathbf{r}_1$ and $\mathbf{r}_2$, which we obtain as eigenvectors of the regularization tensor [54, 216]. It also allows treating both directions independently which is reflected in the use of two separate penalizer functions, which we choose to be the Perona-Malik penalizer ($\Psi_1 = \Psi_p$) and the Charbonnier penalizer ($\Psi_2 = \Psi_c$), respectively.

Regularization Term (Flow) In the case of the order-adaptive regularizer, we choose the non-local selection scheme introduced in Section 4.2.4, which has shown to perform equally well in scenes with fronto-parallel and affine motion. The regularizer combines a first and second-order smoothness term with a locally varying weight. Its general form is given by

$$R_{\text{oar}}(\mathbf{w}, o) = \inf_{\mathbf{a},\mathbf{b}} \left\{\bar{o}\cdot S_1(\mathbf{w}) + (1-\bar{o})\cdot(S_2(\mathbf{w}, \mathbf{a}, \mathbf{b}) + \tau) + \lambda \cdot S_{\text{aux}}(\mathbf{a}, \mathbf{b}) + \theta \cdot \phi(o)\right\}, \tag{5.10}$$

where S_1 is a first-order regularizer, S_2 and S_{aux} form the coupling and smoothness term of a second-order regularizer, respectively, and ϕ is the associated selection term function. In order to avoid over-fitting the data by only selecting the less restrictive second-order regularizer, an activa-

tion cost τ is introduced in the coupling term. Moreover, the selection process is rendered more robust by integrating the order weights o within a rectangular shaped neighborhood $\mathcal{N}(\mathbf{x})$ via

$$\bar{o}(\mathbf{x}) = \frac{1}{|\mathcal{N}(\mathbf{x})|} \int_{\mathcal{N}(\mathbf{x})} o(\mathbf{y})\, d\mathbf{y}\,, \tag{5.11}$$

where $|\mathcal{N}(\mathbf{x})|$ is the size of the neighborhood.

Let us now detail the employed first and second-order smoothness terms. While the first-order smoothness term is given by the anisotropic model [216]

$$S_1(\mathbf{w}) = \sum_{l=1}^{2} \Psi_l\left(\left(\mathbf{r}_l^\top \nabla u\right)^2 + \left(\mathbf{r}_l^\top \nabla v\right)^2\right), \tag{5.12}$$

the second order coupling approach is given by [75, 10]

$$S_2(\mathbf{w}, \mathbf{a}, \mathbf{b}) = \sum_{l=1}^{2} \Psi_l\left(\left(\mathbf{r}_l^\top (\nabla u - \mathbf{a})\right)^2 + \left(\mathbf{r}_l^\top (\nabla v - \mathbf{b})\right)^2\right), \tag{5.13}$$

which couples the flow gradients to the auxiliary functions $\mathbf{a}$ and $\mathbf{b}$ and

$$S_{\text{aux}}(\mathbf{a}, \mathbf{b}) = \sum_{l=1}^{2} \Psi_l\left(\sum_{m=1}^{2}\left(\mathbf{r}_m^\top \mathcal{J}\mathbf{a}\, \mathbf{r}_l\right)^2 + \left(\mathbf{r}_m^\top \mathcal{J}\mathbf{b}\, \mathbf{r}_l\right)^2\right), \tag{5.14}$$

that enforces smoothness on these auxiliary functions via penalizing their Jacobians $\mathcal{J}\mathbf{a}$ and $\mathcal{J}\mathbf{b}$. In this context, the weight λ determines the amount of smoothness and both the directions $\mathbf{r}_1, \mathbf{r}_2$ and the penalizer functions Ψ_1, Ψ_2 are defined as in case of the illumination coefficient regularizer, i.e., as the Perona-Malik penalizer ($\Psi_1 = \Psi_p$) and the Charbonnier penalizer ($\Psi_2 = \Psi_c$), respectively. Finally, the selection term ϕ is given by

$$\phi(o) = (1 - o)\ln(1 - o) - o\ln(o)\,, \tag{5.15}$$

which leads to the order adaptive selection via a sigmoid function based on the local energy differences of S_1 and S_2, where the slope is determined by the factor θ of the selection term. Subsection 4.2.4 includes a detailed derivation of this selection term.

5.7 Minimization

Regarding the minimization of the variational refinement model, we proceed as in Chapter 4 with some slight changes. In contrast to the original approach, we do not estimate the optical flow from scratch which would require to start at a very coarse resolution. Instead, we aim at refining the initial flow field provided by the preceding pipeline steps such that we can benefit from a typically somewhat decent initialization. While the classical optical flow pipeline [143] only operates on the finest resolution, to obtain sub-pixel precision [20, 50, 85, 120, 143], we also do not follow this other extreme. Considering the problem that depending on the matching strategy, the initial matches

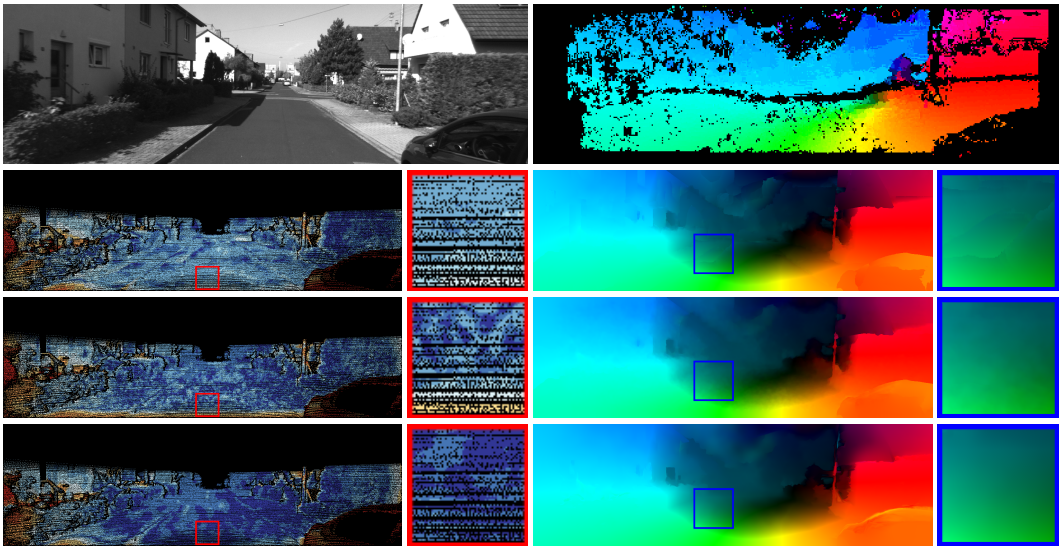


Figure 5.2: Training sequence #28 of KITTI 2012 [65]. *First row*: Reference image, CPM matches. *From left to right*: Error and motion field visualization. *Second row*: Inpainted CPM matches (BP: 7.86%). *Third row*: EpicFlow refinement (BP: 10.80%). *Fourth row*: Proposed refinement (BP: 7.03%).

can be off by several pixels – in particular if matching approaches operate with reduced image resolution, e.g., [50] – we propose a compromise instead: a reduced coarse-to-fine scheme that starts the refinement at an intermediate level and hence still allows for sufficient corrections compared to a refinement on a single scale.

5.8 Evaluation

Evaluation Setup The matching and the inpainting is performed using the publicly available code, provided by the respective authors. Thereby, we set all the parameters to the provided default values. In the case of our model, we set most parameters fixed γ, ζ, τ, θ, and only optimize the three smoothness weights α, β, λ using downhill simplex on the provided training data [14], see Section B.3. To avoid a bias towards a specific matching approach (DeepMatching, CPM, DiscreteFlow), we compute an individual set of these three parameters for each of the three methods, respectively. Moreover, we set the minimum average benefit τ of the order-adaptive regularizer fixed, such that the regularization order is not implicitly learned from the training data for each benchmark in advance (as in Section 4.2), but purely determined online, i.e., during the estimation.

Order-Adaptive Refinement In our first experiment, we demonstrate the benefit of our order-adaptive refinement strategy compared to the first-order refinement of the EpicFlow model in case of highly non-fronto-parallel motion. Therefore, we depicted the results for a sequence of the KITTI 2012 benchmark [65] in Figure 5.2. Taking a look at the second row, which shows the inpainted CPM matches before variational refinement, one can see in the error visualization that the affine inpainting did an excellent job at the bottom boundary, e.g., red framed region. When applying the refinement with the first-order EpicFlow model (third row) this inpainted

Table 5.1: Results for the KITTI 2012 [65], the KITTI 2015 [119] and the MPI Sintel [44] training data sets. The listed error measures are the average endpoint error (AEE) and the percentage of erroneous pixels (BP) with a threshold of 3px.

	KITTI 2012				KITTI 2015				Sintel
	non-occluded		all		non-occluded		all		clean
	AEE	BP (%)	AEE	BP (%)	AEE	BP (%)	AEE	BP (%)	AEE
no refinement									
DeepMatches	1.86 px	11.39 %	3.52 px	18.92 %	5.12 px	24.74 %	9.37 px	31.96 %	2.68 px
DiscreteFlow	**1.36** px	7.25 %	3.06 px	16.02 %	**3.14** px	**15.31** %	**6.67** px	24.37 %	2.21 px
CPM	1.43 px	**6.23** %	**2.99** px	**10.99** %	3.66 px	16.43 %	7.77 px	**23.36** %	**2.19** px
EpicFlow refinement									
DeepMatches	1.42 px	7.64 %	3.24 px	16.24 %	4.71 px	20.06 %	9.18 px	28.38 %	2.27 px
DiscreteFlow	**1.17** px	5.70 %	**2.97** px	14.89 %	**2.94** px	**13.62** %	**6.68** px	23.14 %	**1.94** px
CPM	1.25 px	**5.37** %	3.00 px	**14.58** %	3.43 px	14.58 %	7.78 px	**22.86** %	2.00 px
our refinement ($\eta = 1.00$)									
DeepMatches	1.32 px	7.65 %	2.83 px	12.82 %	4.59 px	19.60 %	8.82 px	26.14 %	2.26 px
DiscreteFlow	**1.08** px	5.80 %	**2.54** px	11.09 %	**2.83** px	**13.03** %	**6.29** px	**20.08** %	**1.91** px
CPM	1.20 px	**5.66** %	2.92 px	**10.19** %	3.85 px	14.11 %	8.88 px	20.57 %	1.99 px
our refinement ($\eta = 0.95$)									
DeepMatches	1.20 px	6.28 %	2.61 px	10.91 %	4.45 px	17.69 %	8.45 px	23.89 %	2.24 px
DiscreteFlow	**1.02** px	**5.05** %	**2.39** px	**9.77** %	**2.79** px	**12.43** %	**5.99** px	**18.56** %	**1.91** px
CPM	1.14 px	5.20 %	2.79 px	9.83 %	3.25 px	13.39 %	7.43 px	19.43 %	2.01 px
our refinement ($\eta = 0.90$)									
DeepMatches	1.16 px	5.67 %	2.52 px	10.06 %	4.32 px	16.25 %	8.25 px	22.33 %	2.23 px
DiscreteFlow	**1.01** px	**4.87** %	**2.34** px	**9.29** %	**2.77** px	**12.16** %	**5.89** px	**18.10** %	**1.94** px
CPM	1.14 px	5.18 %	2.78 px	9.68 %	3.24 px	13.25 %	7.36 px	19.21 %	2.04 px

region deteriorates, but small displacements located at the image center improve. In contrast, our order-adaptive refinement strategy (fourth row) improves both the inpainted areas as well as the small displacements located at the image center. This finding is also reflected in the error measures of the entire KITTI 2012 benchmark for the CPM matches that are listed in Table 5.1. While the BP error increases after the EpicFlow refinement from 10.99% to 14.58 %, it decreases to 9.68 % with our new refinement scheme.

Reduced Coarse-to-fine Scheme In our second experiment, we investigate the proposed reduced coarse-to-fine scheme. Using 10 resolution levels, we thereby compare three different settings for the downsampling parameter η which correspond to three different initial scales: $\eta = 1.0$ (no coarse-to-fine scheme, i.e., full resolution), $\eta = 0.95$ ($0.63\times$full resolution), and $\eta = 0.90$ ($0.39\times$full resolution). Table 5.1 lists the outcome. Here one can see, that in case of the KITTI benchmarks the results benefit significantly from the reduced coarse-to-fine scheme. This improvement is due to the fact that the smaller initial resolution allows for greater corrections. In contrast, one cannot observe such an improvement for the MPI Sintel benchmark. This probably results from the fact that the errors in the inpainted motion field are either small enough to be corrected at a finer resolution or too large to be corrected by the refinement scheme. Regarding the setting with the coarsest resolution the DiscreteFlow matches and the CPM matches turn out to produce slightly worse results, which probably results from the fact that very small structures that are present in the full resolution initialization are lost on the coarser resolution levels.

Table 5.2: Results for the KITTI 2012 [65] and KITTI 2015 [119] test data set. Table shows the top non-anonymous pure optical flow methods at time of submission (Apr. 2017), excluding methods that rely on additional information, such as stereo images, extra time-frames, semantic information or assume an underlying epipolar geometry, and related methods. Our approach (DF+OIR) and the original DiscreteFlow approach are highlighted in red. The methods presented in Section 4.1 (SODA-Flow) and Section 4.2 (OAR-Flow) are highlighted in blue.

KITTI 2012	Out-Noc	Out-All	Avg-Noc	Avg-All
ImpPB+SPCI [154]	4.65 %	13.47 %	1.1 px	2.9 px
FlowNet2 [89]	4.82 %	8.80 %	1.0 px	1.8 px
FlowFieldCNN [22]	4.89 %	13.01 %	1.2 px	3.0 px
RicFlow [84]	4.96 %	13.04 %	1.3 px	3.2 px
FlowFields+ [21]	5.06 %	13.14 %	1.2 px	3.0 px
DF+OIR [8]	**5.17 %**	**10.43 %**	**1.1 px**	**2.9 px**
PatchBatch [61]	5.29 %	14.17 %	1.3 px	3.3 px
SODA-Flow [10]	5.57 %	10.71 %	1.3 px	2.8 px
OAR-Flow [9]	5.69 %	10.72 %	1.4 px	2.8 px
DDF [73]	5.73 %	14.18 %	1.4 px	3.4 px
PH-Flow [202]	5.76 %	10.57 %	1.3 px	2.9 px
FlowFields [20]	5.77 %	14.01 %	1.4 px	3.5 px
CPM-Flow [85]	5.79 %	13.70 %	1.3 px	3.2 px
NLTGV-SC [137]	5.93 %	11.96 %	1.6 px	3.8 px
DDS-DF [184]	6.03 %	13.08 %	1.6 px	4.2 px
TGV2ADCSIFT [33]	6.20 %	15.15 %	1.5 px	4.5 px
S2F-IF [203]	6.20 %	15.68 %	1.4 px	3.5 px
DiscreteFlow [120]	6.23 %	16.63 %	1.3 px	3.6 px
BTF-ILLUM [54]	6.52 %	11.03 %	1.5 px	2.8 px
EpicFlow [143]	7.88 %	17.08 %	1.5 px	3.8 px

KITTI 2015	Fl-bg	Fl-fg	Fl-all
FlowNet2 [89]	10.75 %	8.75 %	10.41 %
DCFlow [198]	13.10 %	23.70 %	14.86 %
SOF [156]	14.63 %	22.83 %	15.99 %
DF+OIR [8]	**15.11 %**	**23.45 %**	**16.50 %**
ImpPB+SPCI [154]	17.25 %	20.44 %	17.78 %
FlowFieldCNN [22]	18.33 %	20.42 %	18.68 %
RicFlow [84]	18.73 %	19.09 %	18.79 %
FlowFields+ [21]	19.51 %	21.26 %	19.80 %
PatchBatch [61]	19.98 %	26.50 %	21.07 %
DDF [73]	20.36 %	25.19 %	21.17 %
SODA-Flow [10]	20.01 %	29.14 %	21.53 %
DiscreteFlow [120]	21.53 %	21.76 %	21.57 %
OAR-Flow [9]	20.62 %	27.67 %	21.79 %
CPM-Flow [85]	22.32 %	22.81 %	22.40 %
FullFlow [50]	23.09 %	24.79 %	23.37 %
SPM-BP [110]	24.06 %	24.97 %	24.21 %
EpicFlow [143]	25.81 %	28.69 %	26.29 %
DeepFlow [191]	27.96 %	31.06 %	28.48 %
HS [164]	39.90 %	51.39 %	41.81 %
DB-TV-L1 [210]	47.52 %	48.27 %	47.64 %

Qualitative Results In the Figures 5.3-5.8, we provide additional qualitative results for sequences of all three considered benchmarks, namely the KITTI 2012 benchmark [65], the KITTI 2015 benchmark [119] and the MPI Sintel benchmark [44]. To emphasize some aspects of our novel refinement strategy, we highlighted specific regions in the images. Figure 5.3 and Figure 5.4, for example, nicely show the benefit of the proposed reduced coarse-to-fine scheme, which allows correcting errors. Figure 5.5 and Figure 5.8 bring out the adaptation to the underlying image structure, as can be seen at the traffic lights and the ear of the villain, respectively. Finally, the benefit of the order-adaptive regularization not only becomes present in ego-motion scenarios, e.g., the boundary areas of the KITTI sequences (Figures 5.3-5.5), but also in case of non-rigid motion, e.g., the shoulder of the character Sintel in Figure 5.7.

Comparison to the Literature Finally, we evaluate our new order-adaptive variational refinement strategy on the withhold test data sets of the KITTI 2012 benchmark [65], the KITTI 2015 benchmark [119] and the MPI Sintel benchmark [44], by uploading the computed flow field to the online evaluation servers. Following the submission policy, we submitted the best performing setting, i.e., the combination of our DiscreteFlow setting (DiscreteFlow matches + filtering + inpainting) with our order-adaptive refinement strategy. For a convenient overview, we provide the results from the time of submission (Apr. 2017) in Table 5.2 and Table 5.3 where we only listed non-anonymous pure optical flow methods that do not rely on additional information, such as stereo images, extra time-frames, semantic information or assume an underlying epipo-

Table 5.3: Results for the MPI Sintel [44] test data set in terms of the average enpoint error (AEE). Top non-anonymous optical flow methods and related methods at time of submission (Apr. 2017). Our approach (DiscreteFlow+OIR) and the original DiscreteFlow approach are highlighted in red. The method presented in Section 4.2 (OAR-Flow) is highlighted in blue.

clean render path	all	matched	unmatched	final render path	all	matched	unmatched
FlowFields+ [21]	3.102 px	0.820 px	21.718 px	DCFlow [198]	5.119 px	2.283 px	28.228 px
CPM2 [111]	3.253 px	0.980 px	21.812 px	FlowFieldsCNN [22]	5.363 px	2.303 px	30.313 px
DiscreteFlow+OIR [8]	**3.331** px	**0.942** px	**22.817** px	S2F-IF [203]	5.417 px	2.549 px	28.795 px
S2F-IF [203]	3.500 px	0.988 px	23.986 px	RicFlow [84]	5.620 px	2.765 px	28.907 px
SPM-BPv2 [110]	3.515 px	1.020 px	23.865 px	FlowFields+ [21]	5.707 px	2.684 px	30.356 px
DCFlow [198]	3.537 px	1.103 px	23.394 px	DeepDiscreteFlow [73]	5.728 px	2.623 px	31.042 px
RicFlow [84]	3.550 px	1.264 px	22.220 px	FlowNet2-ft-sintel [89]	5.739 px	2.752 px	30.108 px
CPM-Flow [85]	3.557 px	1.189 px	22.889 px	FlowFields [20]	5.810 px	2.621 px	31.799 px
DiscreteFlow [120]	3.567 px	1.108 px	23.626 px	SPM-BPv2 [110]	5.812 px	2.754 px	30.743 px
FullFlow [50]	3.601 px	1.296 px	22.424 px	DiscreteFlow+OIR [8]	**5.862** px	**2.864** px	**30.303** px
PatchBatch+Inter [154]	3.624 px	1.324 px	22.397 px	FullFlow [50]	5.895 px	2.838 px	30.793 px
FlowFields [20]	3.748 px	1.056 px	25.700 px	CPM-Flow [85]	5.960 px	2.990 px	30.177 px
FlowFieldsCNN [22]	3.778 px	0.996 px	26.469 px	FlowNet2 [89]	6.016 px	2.977 px	30.807 px
DeepDiscreteFlow [73]	3.863 px	1.296 px	24.820 px	GlobalPatchCollider [180]	6.040 px	2.938 px	31.309 px
FlowNet2 [89]	3.959 px	1.468 px	24.294 px	DiscreteFlow [120]	6.077 px	2.937 px	31.685 px
EpicFlow [143]	4.115 px	1.360 px	26.595 px	EpicFlow [143]	6.285 px	3.060 px	32.564 px
OAR-Flow [9]	6.227 px	2.760 px	34.455 px	OAR-Flow [9]	8.179 px	4.578 px	37.525 px

lar geometry. Moreover, we added results from EpicFlow [143] and OAR-Flow [9], if not already present among the list of the best results, since these methods rely on the standard pipeline and our order-adaptive regularization from Section 4.2, respectively. As one can see, our variational refinement not only improves the results compared to the original DiscreteFlow approach (DiscreteFlow matches + filtering + inpainting + EpicFlow refinement) and the standard EpicFlow pipeline (DeepMatches + filtering + inpainting + EpicFlow refinement), it also outperforms recent purely variational methods with full coarse-to-fine schemes such as our double anisotropic second-order approach from Section 4.1 (SODA-Flow), our order-adaptive approach from Section 4.2, and our baseline approach [54] with the illumination-aware data term from Subsection 4.1.2 (BTF-Illum). Moreover, with Rank 6 (KITTI 2012), Rank 4 (KITTI 2015), and Ranks 3 and 10 (MPI Sintel) in the above Tables, the novel refinement approach offers a favorable performance in all benchmarks. These results demonstrate that combining good initial matches with a sophisticated variational refinement allows to further improve the results by combining the advantages of both techniques.

RUNTIME For a color image pair of size 1242×375 (KITTI 2015 [119]), our C/C++ implementation of the variational refinement step running on a single core with 3.40 GHz (Intel Core i7-2600 CPU) requires about 35s ($\eta = 0.90$), 50s ($\eta = 0.95$), and 70s ($\eta = 1.00$). Consequently, our reduced coarse-to-fine approach not only allows to improve the estimation accuracy but also to reduce the runtime. Furthermore, the overall runtime of the presented approach is dependent on the previous steps of the pipeline. Using the same hardware the previous pipeline steps sum up to 12s (CPM), 80s (Deepmatches) and 120s (DiscreteFlow).

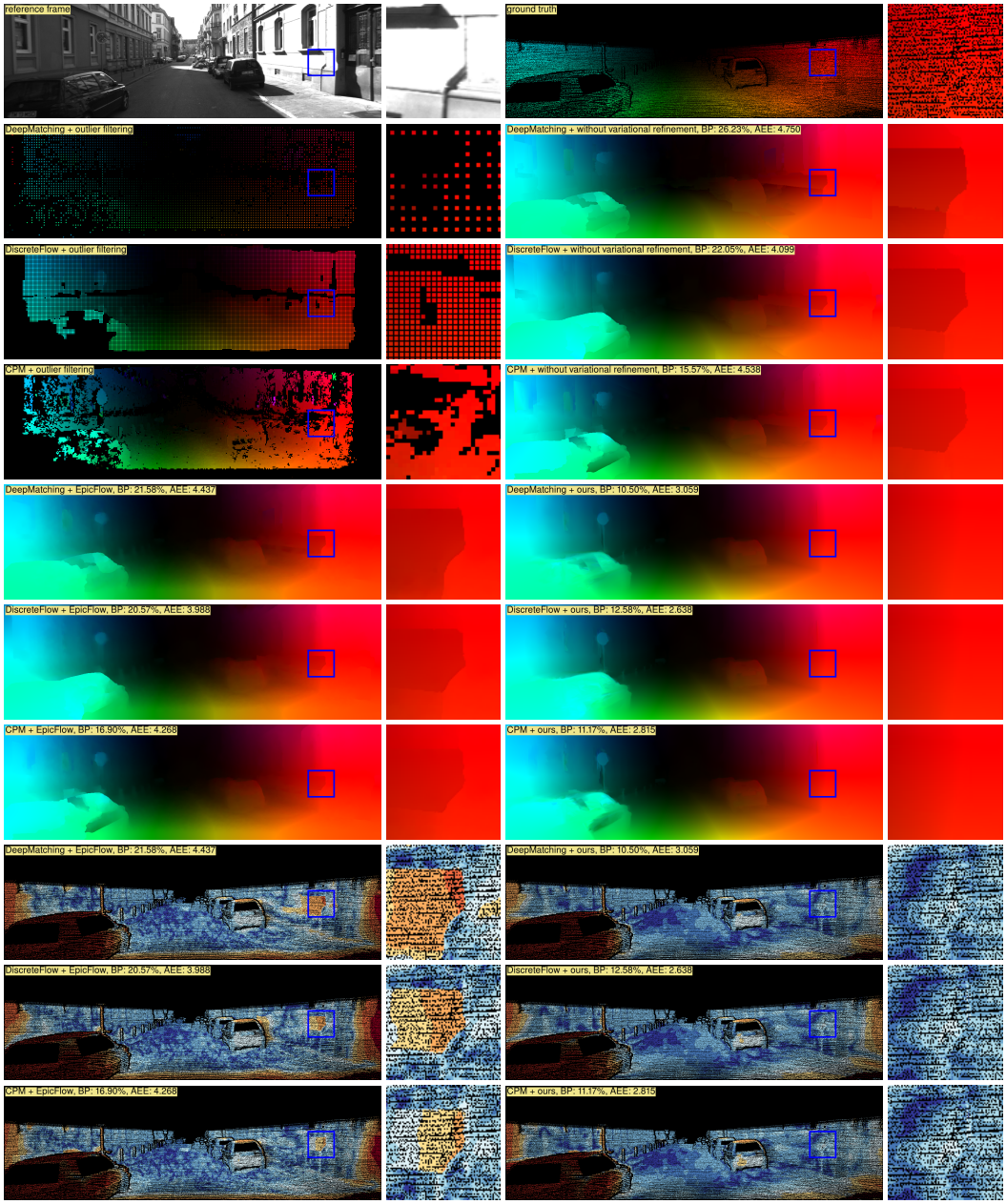


Figure 5.3: Results for the sequence #15 of the KITTI 2012 benchmark [65]. *First row*: Reference frame and ground truth. *Second to fourth row*: Outlier-filtered matches and inpainted matches (DeepMatching, DiscreteFlow, CPM). *Fifth to seventh row*: Flow field visualization of the EpicFlow refinement and proposed refinement (DeepMatching, DiscreteFlow, CPM). *Eighth to tenth row*: Error visualization of the EpicFlow refinement and our refinement (DeepMatching, DiscreteFlow, CPM).

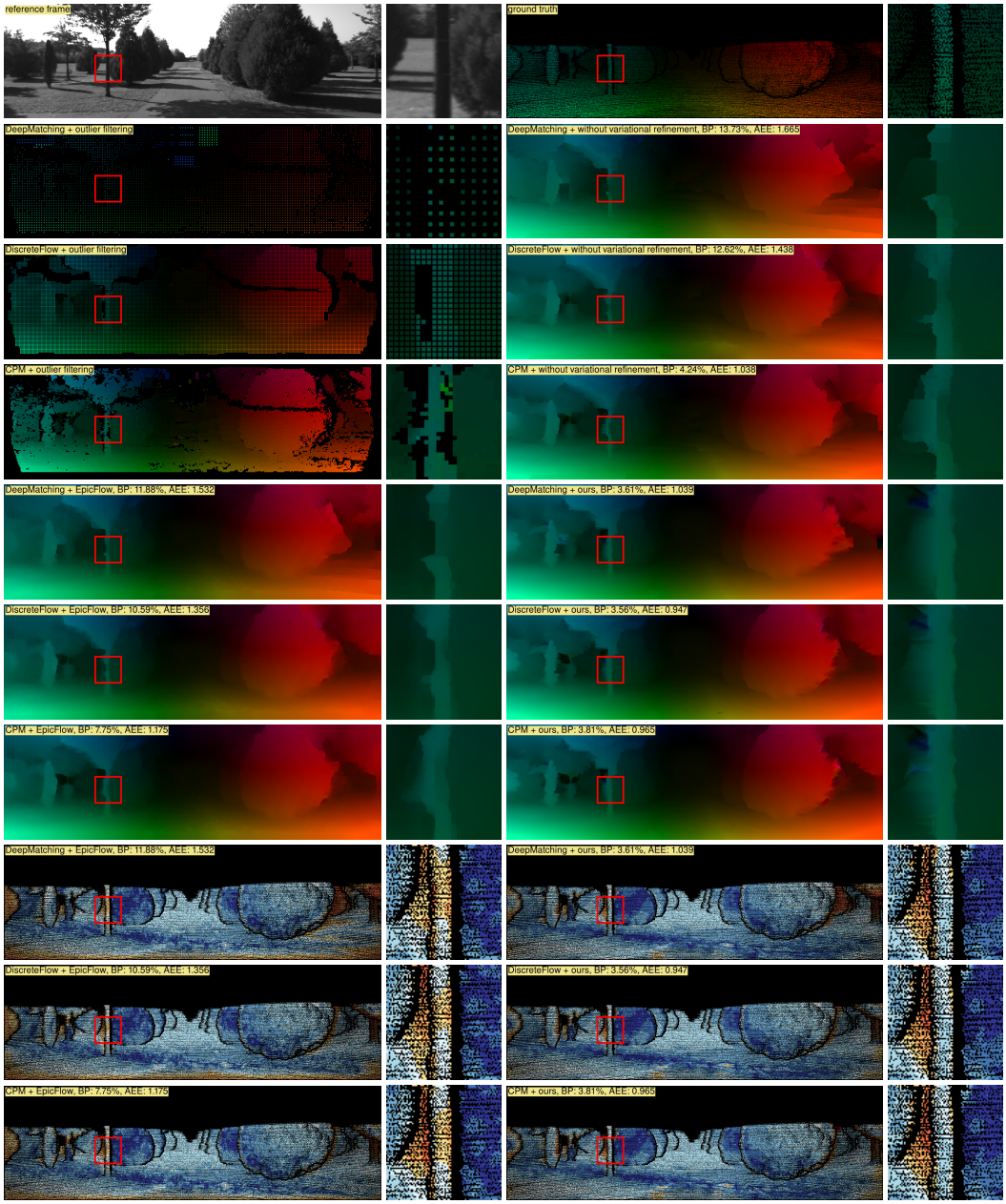


Figure 5.4: Results for the sequence #9 of the KITTI 2012 benchmark [65]. *First row*: Reference frame and ground truth. *Second to fourth row*: Outlier-filtered matches and inpainted matches (DeepMatching, DiscreteFlow, CPM). *Fifth to seventh row*: Flow field visualization of the EpicFlow refinement and proposed refinement (DeepMatching, DiscreteFlow, CPM). *Eighth to tenth row*: Error visualization of the EpicFlow refinement and our refinement (DeepMatching, DiscreteFlow, CPM).

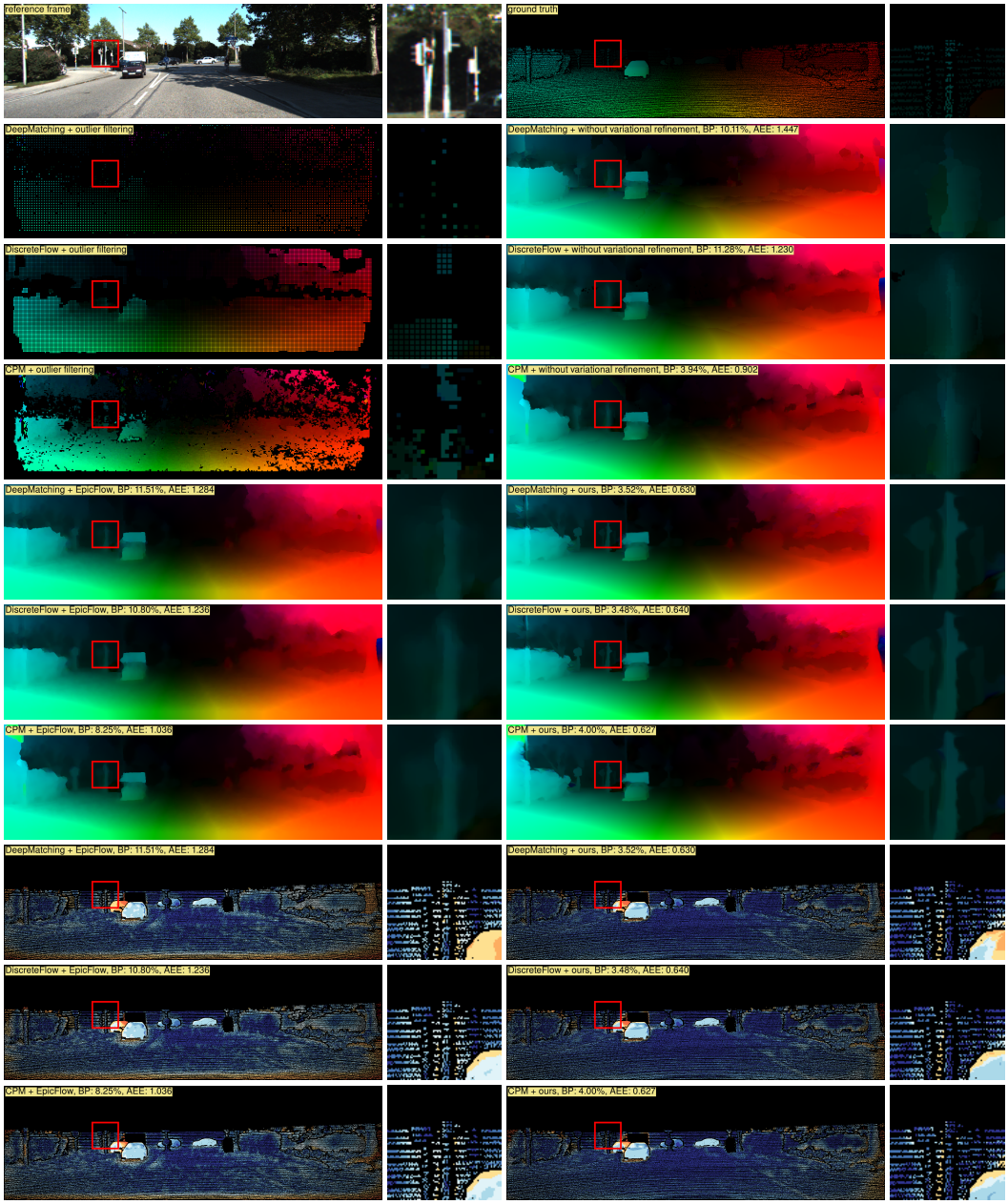


Figure 5.5: Results for the sequence #07 of the KITTI 2015 benchmark [119]. *First row*: Reference frame and ground truth. *Second to fourth row*: Outlier-filtered matches and inpainted matches (DeepMatching, DiscreteFlow, CPM). *Fifth to seventh row*: Flow field visualization of the EpicFlow refinement and proposed refinement (DeepMatching, DiscreteFlow, CPM). *Eighth to tenth row*: Error visualization of the EpicFlow refinement and our refinement (DeepMatching, DiscreteFlow, CPM).

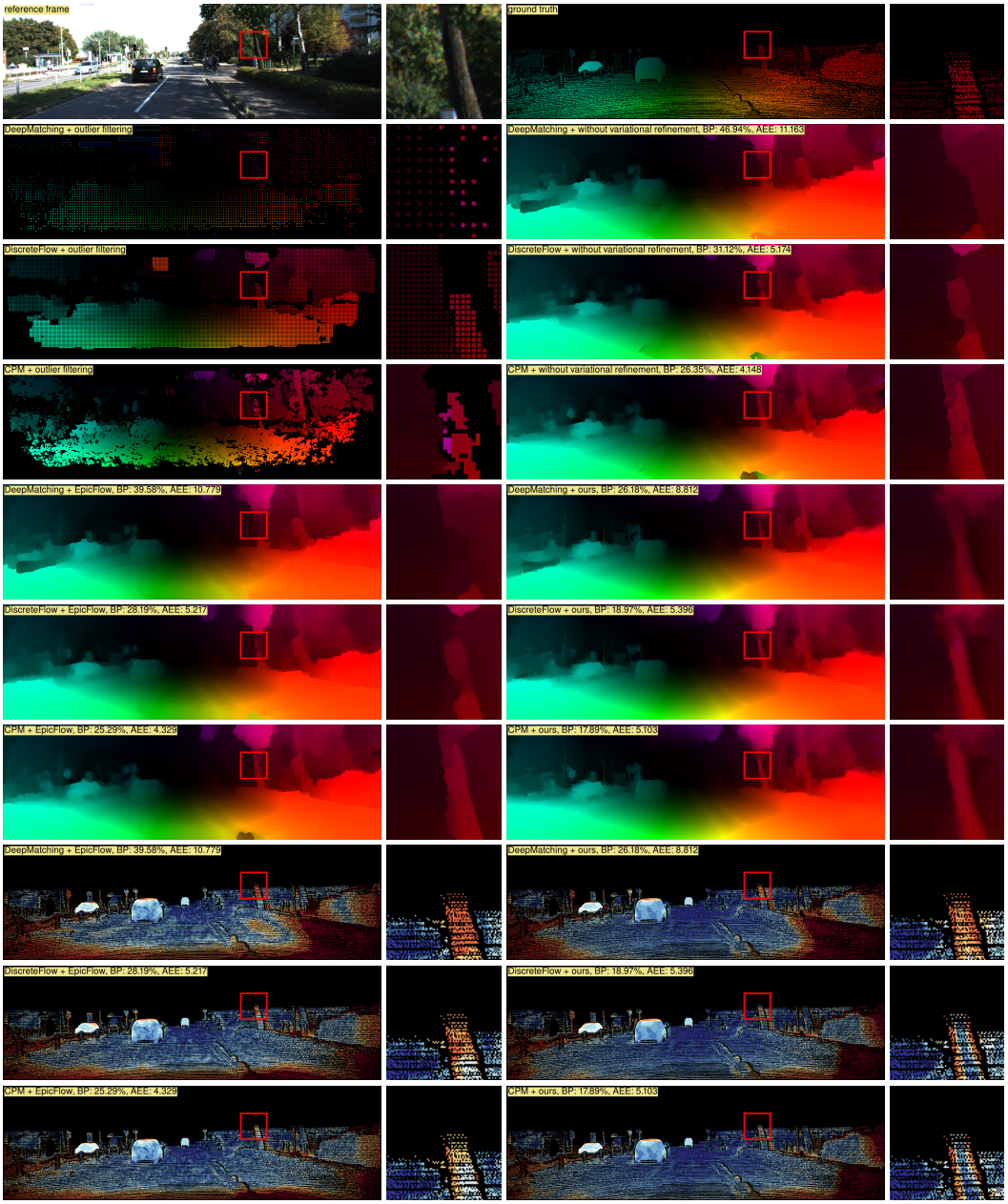


Figure 5.6: Results for the sequence #31 of the KITTI 2015 benchmark [119]. *First row*: Reference frame and ground truth. *Second to fourth row*: Outlier-filtered matches and inpainted matches (DeepMatching, DiscreteFlow, CPM). *Fifth to seventh row*: Flow field visualization of the EpicFlow refinement and proposed refinement (DeepMatching, DiscreteFlow, CPM). *Eighth to tenth row*: Error visualization of the EpicFlow refinement and our refinement (DeepMatching, DiscreteFlow, CPM).

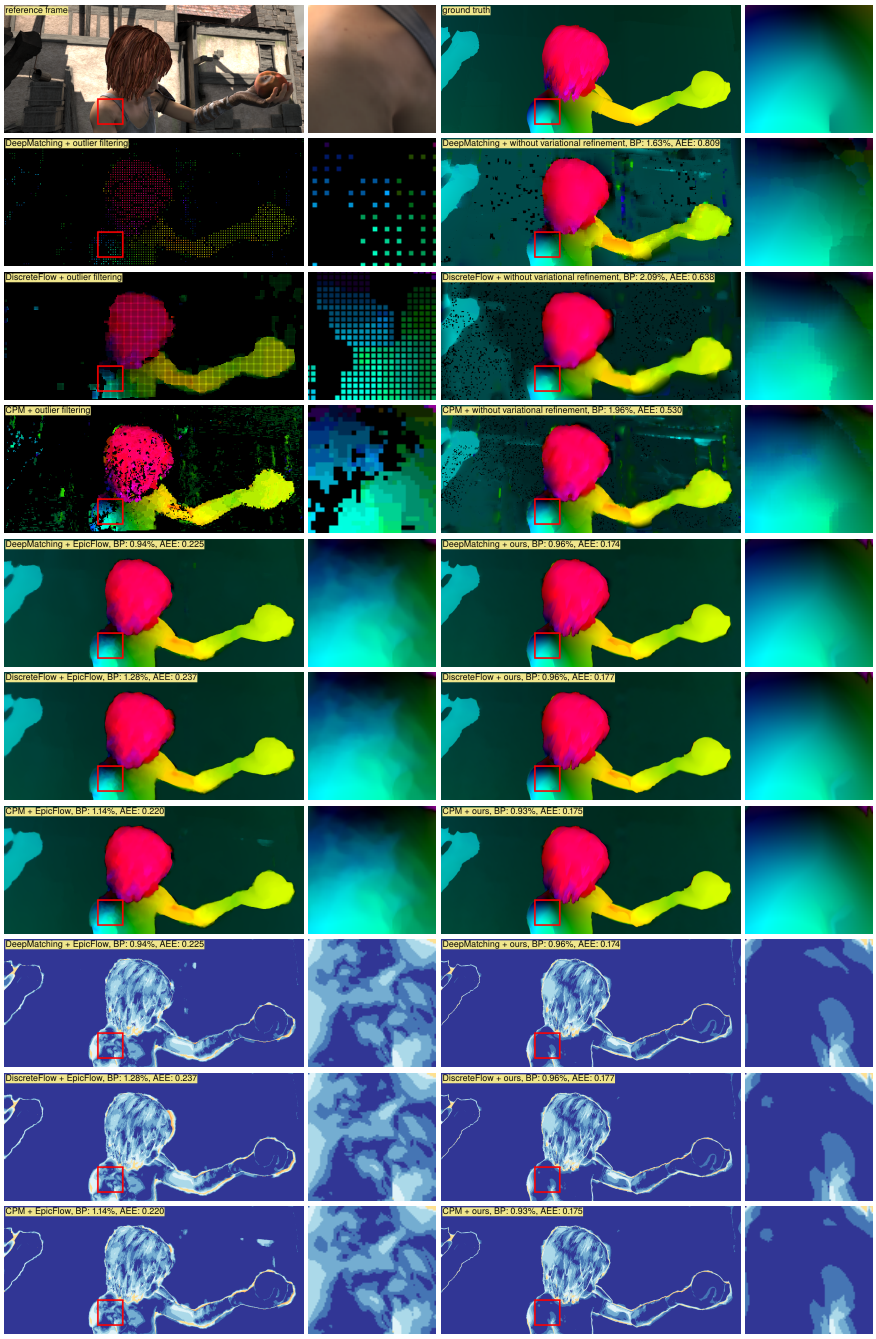


Figure 5.7: Results for sequence #20 (alley_1) of the MPI Sintel benchmark [44]. *First row*: Reference frame and ground truth. *Second to fourth row*: Outlier-filtered matches and inpainted matches (DeepMatching, DiscreteFlow, CPM). *Fifth to seventh row*: Flow field visualization of the EpicFlow refinement and proposed refinement (DeepMatching, DiscreteFlow, CPM). *Eighth to tenth row*: Error visualization of the EpicFlow refinement and our refinement (DeepMatching, DiscreteFlow, CPM).

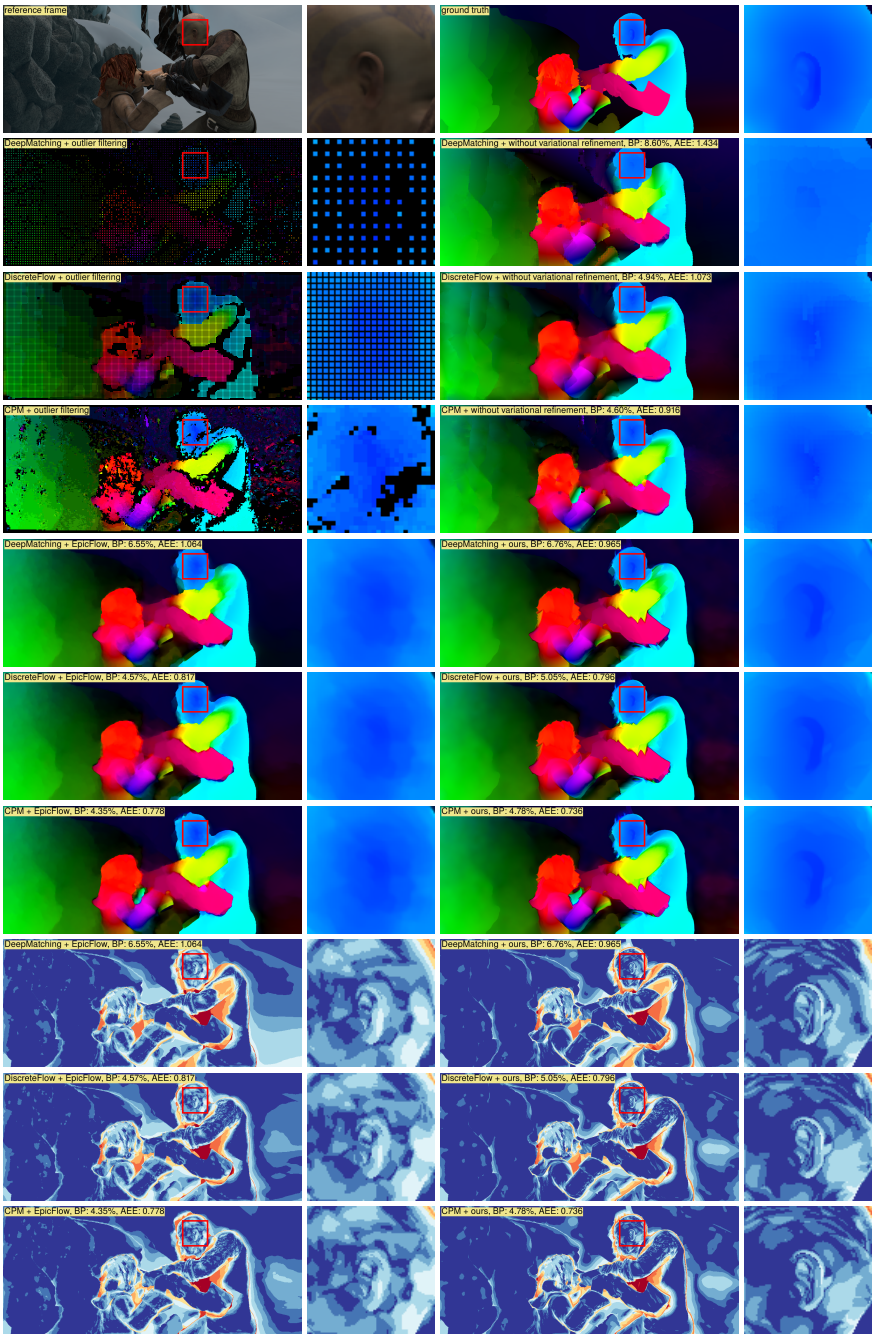


Figure 5.8: Results for sequence #02 (ambush_5) of the MPI Sintel benchmark [44]. *First row*: Reference frame and ground truth. *Second to fourth row*: Outlier-filtered matches and inpainted matches (DeepMatching, DiscreteFlow, CPM). *Fifth to seventh row*: Flow field visualization of the EpicFlow refinement and proposed refinement (DeepMatching, DiscreteFlow, CPM). *Eighth to tenth row*: Error visualization of the EpicFlow refinement and our refinement (DeepMatching, DiscreteFlow, CPM).

5.9 Limitations

Our approach not only enables an accurate refinement of flow fields but also is capable of correcting errors. Naturally, this error correction capability is limited when it comes to erroneous flow vectors of small objects that undergo a large displacement. Such errors cannot be corrected if no correct matches are captured during the matching phase. Further, we can observe another limiting scenario at motion boundaries between foreground objects and homogeneous background regions. In this case, it can appear that despite the edge enhancing and edge-preserving penalizer functions, the smoothness term over-smooths the edge. Additional segmentation information may help in this context.

5.10 Conclusion

In this chapter, we proposed a new variational refinement strategy for pipeline based optical flow estimation. By combining an illumination-aware data term, that can keep up with many feature descriptors regarding their robustness under affine changes, with our new order-adaptive regularization strategy from Section 4.2, that locally selects between first and second-order regularization, we build a new variational model that unifies modern concepts from the field of purely variational optical estimation. Furthermore, we not only came up with a new model but also proposed a reduced coarse-to-fine scheme that starts the computation at an intermediate level. This compromise enabled our new refinement strategy to benefit from a good initialization while still being able to correct errors. Finally, consistently good results on recent optical flow benchmarks showed that our new variational refinement strategy not only allows to improve outcomes compared to traditional refinement schemes but also that it allows outperforming purely variational methods.

6 Multi-Frame Motion Estimation

The problem of motion estimation is defined as the computation of the inter-frame displacement field between consecutive image frames. So far, as many recent methods [84, 87, 8, 163, 198], we limited ourselves to the use of only two input frames to solve this task. While this choice allows obtaining excellent results in most cases, it does not allow to exploit any information on temporal coherence. In particular reasoning in the context of occlusions is solely based on regularization. However, taking multiple input frames into account enables us to leverage additional information, which could help to overcome this limitation. In order to achieve this, motion models are required that relate the sought displacement vector field to motion estimates from the past. While simple models based on a temporally constant flow can be a valid choice in case of sufficiently small motion [91], more complex models are required in general scenarios with fast and non-rigidly moving objects. Unfortunately, as observed in [64, 179], finding such models is a highly non-trivial task. Thus, recent multi-frame methods assume a scenario of mostly rigid scenes to use temporal information [6, 196]. This assumption, however, requires a sufficient amount of ego-motion and only allows to exploit temporal information in rigid parts of the scene.

In this chapter we present two new methods that exploit additional temporal information from multiple input frames. While our first method builds upon the mentioned rigid motion model, our second method does not rely on any specific motion model. In contrast, we propose to learn a motion model, what allows us to overcome certain limitations imposed by the rigid motion model. Main parts of this chapter are based on the work published in [2, 6].

6.1 Rigid Motion Model

To develop our first multi-frame approach that allows extracting additional temporal information based on the rigid motion assumption, we build upon a pipeline approach as introduced in the previous Chapter 5. In particular, we extend the matching step of the pipeline to include additional structure matches obtained by solving a multi-frame structure-from-motion (SfM) problem. To this end, we create a structure matching algorithm that relies on a PatchMatch-like [25] optimization. Hence, we not only detail on pure optical flow approaches but also point out approaches related to PatchMatch based structure estimation within the related work section.

6.1.1 Related Work

Rigid Motion Actually, no multi-frame setting is required to employ the rigid motion model. Since it allows to reduce the 2D search space of unconstrained motion to a 1D search space

along epipolar lines, it can already model the estimation more accurate and robust in a two-frame setting on condition that the present motion fulfills the underlying rigid motion assumption.

Hence, approaches exist that enforce the rigid motion assumption geometrically in terms of an epipolar constraint in the standard two frame scenario [129, 170, 200, 201]. However, if this assumption is forced to hold for the entire scene, as proposed by Oisel et al. [129] and Yamaguchi et al. [200, 201], the approach is only applicable to entirely rigid scenes, e.g., to those of the KITTI 2012 benchmark [65]. Although this problem can be slightly alleviated by soft constraints as proposed by Valgaerts et al. [169, 170], results for non-rigid scenes are typically not good. Hence, Wedel et al. [182] suggested to turn off the epipolar constraint for sequences with independent object motion. This modification, however, does not allow to exploit rigid body priors at all in the standard optical flow setting, a setting with camera ego motion and independent object motion. Consequently, Gerlich and Eriksson [67] presented a more advanced approach that segments the scene into different regions with independent rigid body motions and assigns motion hypothesis in terms of fundamental matrices to them. While this strategy allows handling automotive scenes with other rigidly moving objects quite well, e.g., sequences similar to the KITTI 2015 benchmark [119], it cannot model any non-rigid motion, e.g., as required for the different characters in the MPI Sintel benchmark [44]. In contrast, our proposed approach can handle non rigid motion by combining information from unconstrained motion estimation and SfM estimation, it is neither restricted to entirely rigid nor to object-wise rigid scenes.

Mostly Rigid Motion Compared to the previously mentioned approach of Gerlich and Eriksson [67], Wulff et al. [196] went a step further. Instead of requiring the scene to be object-wise rigid, they assume the scene to be mostly rigid. To this end, they suggested an iterative model that segments the scene into foreground and background using semantic information as well as motion and structure cues, while estimating the background motion with a dedicated stereo algorithm. In this context, the use of such a rigid motion model in terms of a stereo algorithm allows them to use an additional preceding image frame to exploit additional temporal information. In contrast to their approach, our method follows a completely different strategy. Instead of relying on the general optical flow method from [120] as initialization and adaptively integrating strong rigidity priors later on in the estimation, our proposed approach aims at integrating such priors already in the estimation of feature matches at the beginning of the flow pipeline – and that without the use of semantic information. Hence, our algorithm is relevant for all methods relying on a suitable initialization. In particular, this not only includes the work of Wulff et al. [196] but also other recent methods such as [87] or [156].

Parametrized Rigid Motion An alternative strategy that recently became very popular is to refrain from using global or object-wise rigidity priors and to model motions that are pixel- or piecewise rigid. Typically this is done through a suitable flow (over-)parametrization [82, 86, 119, 128, 176, 202]. For instance, Hornaček et al. [82] proposed a 9 DoF flow parametrization that models a locally rigid motion of planes. Similar, Yang et al. [202] and Hur and Roth [86, 87] suggested approaches that use a spatially coherent 8 DoF homography based on superpixels. In contrast to those methods, our proposed approach does not explicitly rely on an over-parametrization. Vice versa, it gains robustness by restricting the search space to 1D when calculating the structure matches. Moreover, it estimates the flow pixel-wise instead of segment-wise. Hence, it is more suitable for general scenes with non-rigid motion and fine motion details.

Semantic Information Another way to improve the accuracy and the robustness of the estimation is to consider semantic information from the scene. For instance, Bai et al. [19] proposed to use instance-level segmentation to identify independently moving traffic participants before computing separate rigid motions for both the background and the participants. Similarly, Hur and Roth [86] make use of a CNN to integrate semantic information into a joint approach for estimating the flow and a temporally consistent semantic segmentation. Furthermore, Sevilla-Lara et al. [156] suggested a layered approach that relies on semantic information when switching between different motion models. Finally, there is also the method of Wulff et al. [196] that uses semantic information to distinguish independently moving objects from the rigid background. While semantic information often improves the results, one typically has to adapt the underlying models to the given domain. As a consequence, such approaches do typically not generalize well across different applications or benchmarks. To avoid this inevitable application-wise adaptation and come up with a generally applicable method, we propose an approach that does not rely on semantic information. To this end, we propose an algorithm that verifies the reliability of structure matches in terms of a consistency check.

PatchMatch Approaches Finally, let us comment on some related work regarding the use of PatchMatch approaches for motion estimation and stereo reconstruction. In the context of unconstrained matching (motion estimation), PatchMatch has been originally proposed by Barnes et al. [25]. Recent developments include the work of Bao et al. [24] that introduces an edge-preserving weighting scheme in the matching process, as well as the approach of Hu et al. [85] that improves accuracy and speed with a hierarchical matching strategy. Moreover, Gadot and Wolf [61] and Bailer et al. [22], have recently shown that feature learning can be beneficial. Despite all the progress, however, none of the above mentioned motion estimation methods includes structure information. In contrast, our proposed approach exploits such information by explicitly using feature matches from a specifically tailored three-view stereo/SfM PatchMatch method.

Also in the stereo/SfM context, there exists a vast literature on PatchMatch algorithms. There, PatchMatch has been first introduced by Bleyer et al. [31] who proposed a plane-fitting variant for the rectified case. Recent developments include the approaches of Shen [157] and Galliani et al. [62] who extended PatchMatch to the non-rectified two-view and multi-view case, respectively; see also [150, 213]. In contrast to all those methods, our proposed approach not only extracts structure information from images. Instead, it combines information from optical flow and structure and is hence also applicable to non-rigid scenes with independently moving objects. Moreover, it relies on a hierarchical optimization [85] which has not been used in the context of PatchMatch stereo so far. Finally, the structure part of our algorithm uses a direct depth-parametrization. These choices, in turn, make both the estimation and the optimization very robust.

6.1.2 Contributions

In this chapter, we propose and investigate an approach that allows exploiting the multi-frame information based on a rigid motion assumption. In this context, our contributions are threefold: (i) First, we introduce a coarse-to-fine multi-frame PatchMatch approach for estimating structure matches (SfM) that combines a depth based parametrization with different temporal selection strategies. While the parametrization models the estimation more robust by reducing the search space, the hierarchical optimization and the temporal selection improve the accuracy. (ii) Second,

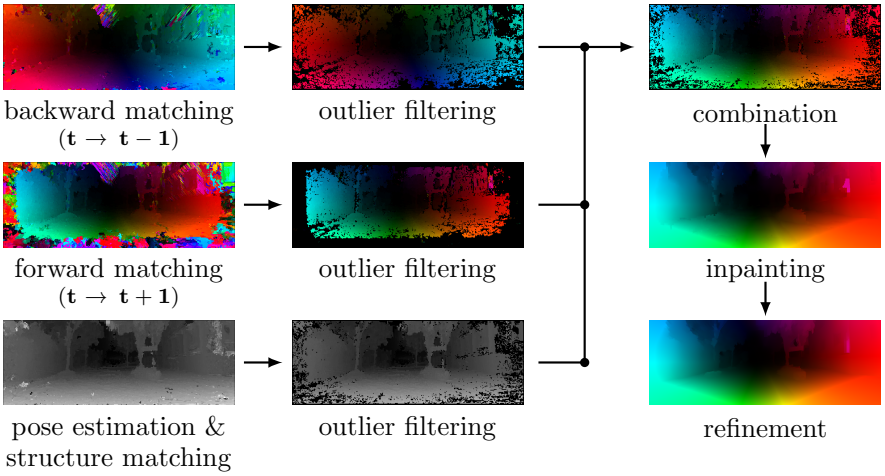


Figure 6.1: Schematic overview over our proposed approach.

we propose a consistency-based selection scheme for combining matches from this structure-based PatchMatch approach and an unconstrained PatchMatch approach. Thereby, the backward flow allows us to identify reliable structure matches, while a robust voting scheme decides on the remaining cases. (iii) Finally, we embed the resulting matches into the optical flow pipeline. By employing recent approaches for interpolation and refinement, our method provides dense results with sub-pixel accuracy. Experiments on all major benchmarks demonstrate the benefits of our novel approach.

6.1.3 Method Overview

Let us start by giving a brief overview of the proposed method. As many modern optical flow techniques it relies on a multi-stage approach as described in the previous chapter [84, 87, 8, 156, 196]. However, in contrast to most of these approaches that typically aim at improving an already given flow field, our method focuses on the generation of an accurate and robust initial flow field itself. To achieve this goal, we integrate structure information into the feature matching process. This integration is motivated by the observation that many sequences contain a significant amount of rigid motion induced by the ego-motion of the camera [196]. Since the underlying stereo geometry constrains this motion, structure information can hence significantly improve the estimation.

In our multi-stage method, we realize this integration by combining two hierarchical feature matching approaches that complement each other: On the one hand, we use a recent two-frame PatchMatch approach for optical flow estimation [85]. This choice allows our method to estimate the unconstrained motion in the scene (forward and backward matches). On the other hand, we rely on a specifically tailored three-frame stereo/SfM PatchMatch approach with preceding pose estimation [122]. This component, in turn, allows our method to compute the rigid motion of the scene induced by the moving camera (structure matches). To discard outliers and combine the remaining matches, we perform a filtering approach for all matches followed by a consistency-based selection scheme. Finally, we inpaint and refine the combined matches using recent methods from the literature [84, 8]. Figure 6.1 gives an overview of the entire approach.

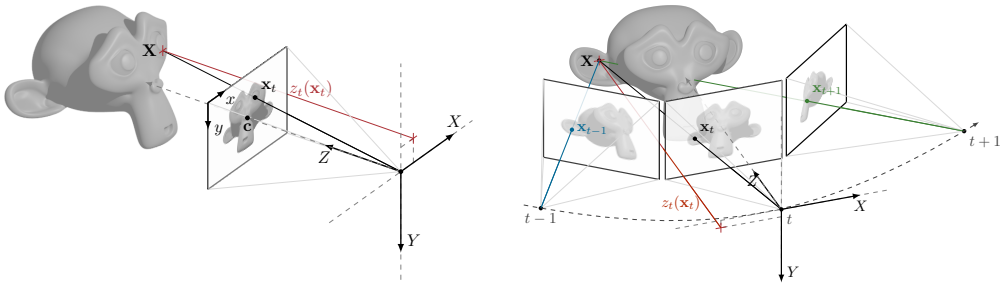


Figure 6.2: *Left:* Illustration of the employed depth parametrization. *Right:* Illustration of corresponding points defined by the image location $\mathbf{x}_t$ and the associated depth value $z_t(\mathbf{x}_t)$. In this case, the 3D point is occluded in one view and could be handled with the idea of temporal selection, i.e. by the view from the other time step.

6.1.4 Structure Matching

In this section, we present our structure matching framework which builds upon the PatchMatch algorithm [25] – a randomized, iterative algorithm for approximate block matching. In this context, we adopt ideas of the recently proposed coarse-to-fine PatchMatch (CPM) approach for optical flow [85] and apply them in the context stereo/SfM estimation. To this end, we rely on a depth-based parametrization [62, 145] that we already used in our variational approach for 3D reconstruction in Chapter 3. This parametrization not only enables the straightforward integration of multiple frames but also allows us to consider the concepts of temporal averaging and temporal selection [98].

6.1.4.1 Depth-Based Parametrization

Let us start by introducing the employed depth-based parametrization. To this end, we assume that all images are captured by a calibrated perspective camera that possibly moves in space, i.e., that we know the corresponding projection matrices $P_t = K\,[R_t|\mathbf{t}_t]$. Here, R_t is a 3×3 rotation matrix and $\mathbf{t}_t$ is a translation 3-vector that together describe the pose of the camera at a particular time step t. Moreover, K denotes the 3×3 intrinsic camera calibration matrix as described in Subsection 2.2.2. Given the projection matrix P_t, a 3D point $\mathbf{X} \in \mathbb{R}^3$ is projected onto a 2D point $\mathbf{x}_t \in \mathbb{R}^2$ on the image plane at time t by $\mathbf{x}_t = \pi(P_t\tilde{\mathbf{X}})$, where the tilde denotes homogeneous coordinates and π maps a homogeneous coordinate $\tilde{\mathbf{x}}_t$ to its Euclidean counterpart $\mathbf{x}_t$. Now, to define our parametrization, we fix a camera at a certain time step t to be the reference frame. This choice allows us to specify a 3D point on the surface $\mathbf{s}_t$ in terms of an image location $\mathbf{x}_t$ and its corresponding depth $z_t(\mathbf{x}_t)$ along the optical axis of the reference camera (see Figure 6.2 left) via the back projection

$$\mathbf{X} = \mathbf{s}_t(\mathbf{x}_t, z_t) = z_t(\mathbf{x}_t) \cdot R_t^{-1}K^{-1}\tilde{\mathbf{x}}_t - R_t^{-1}\mathbf{t}_t\,, \tag{6.1}$$

For the sake of clarity we drop the subscript t in case we refer to the reference view, i.e., $\mathbf{x} := \mathbf{x}_t$, $z := z_t$, and $\mathbf{s} := \mathbf{s}_t$. This depth parametrization enables us to describe correspondences throughout multiple images with a single unknown, the depth z, by projecting onto the respective

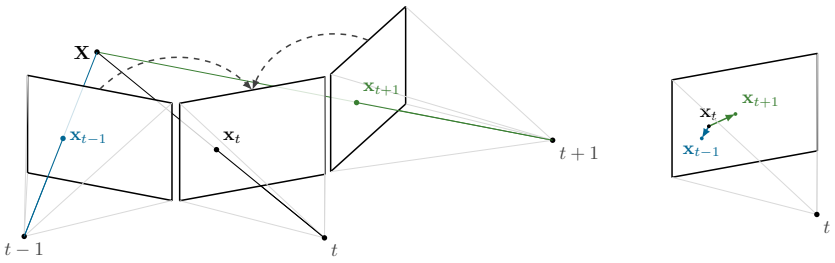


Figure 6.3: Illustration showing the conversion procedure from a 3D point to the displacement vectors w.r.t. to the forward frame $t+1$ and backward frame $t-1$.

image planes using the corresponding projection matrices (see Figure 6.2 right). Finally, given three frames at time $t-1$, t, and $t+1$ as well as the projection matrices P_{t+1}, P_t, and P_{t-1}, one can directly convert the estimated depth values of the reference camera at time t to the corresponding displacement vectors w.r.t. the forward frame $t+1$ and the backward frame $t-1$ (see Figure 6.3)

$$\mathbf{w}_{\text{st, fw}}(\mathbf{x}, z) = \pi(P_{t+1}\tilde{\mathbf{s}}(\mathbf{x}, z)) - \pi(P_t\tilde{\mathbf{s}}(\mathbf{x}, z))\,, \tag{6.2}$$

$$\mathbf{w}_{\text{st, bw}}(\mathbf{x}, z) = \pi(P_{t-1}\tilde{\mathbf{s}}(\mathbf{x}, z)) - \pi(P_t\tilde{\mathbf{s}}(\mathbf{x}, z))\,. \tag{6.3}$$

6.1.4.2 Hierarchical Matching

With the depth parametrization at hand, we now turn to the actual matching to determine z. While applying the classical PatchMatch approach [25] directly to the problem yields noisy results due to non-existent explicit regularization, we resort to the idea of integrating a hierarchical coarse-to-fine scheme, which has shown to be less prone to noise in the context of motion estimation [85].

As in [85] we do not estimate the unknowns for all pixel locations, but for multiple collections of n_s seeds $\mathcal{S}^l = \{s_m^l | m \in \{1, \ldots, n_s\}\}$ that we define on each resolution level $l \in \{0, 1, \ldots, k-1\}$ of the coarse-to-fine pyramid. While the number of seeds remains the same for each resolution level, their spatial locations are given by

$$\mathbf{x}(s_m^l) = \lfloor \eta \cdot \mathbf{x}(s_m^{l-1}) \rceil \quad \text{for} \quad l \geq 1\,, \tag{6.4}$$

where $\lfloor \cdot \rceil$ is a function that returns the nearest integer value, and $\eta = 0.5$ is the employed downsampling factor between two consecutive pyramid levels. Furthermore, the locations for $l = 0$ (full image resolution) are located at the cross points of a regular image grid with a spacing of 3 pixels and come with the default neighborhood system, defined via the spatial adjacency. In addition, these neighborhood relations remain fixed throughout the coarse-to-fine pyramid, also for seeds whose locations coincide on lower resolution levels.

We then perform the matching in the traditional coarse-to-fine manner: Starting at the coarsest resolution, we process each level by iteratively performing a random search and a neighborhood propagation as in [25]. While the coarsest level uses a random initialization of the unknown depth, the subsequent levels are initialized with the depth values of the corresponding seeds of the next coarser level. Furthermore, the search radius for the random sampling is reduced exponentially

throughout the coarse-to-fine pyramid, such that the random search is restricted to values near the current best depth estimate.

6.1.4.3 Cost Computation and Temporal Averaging / Selection

Since we consider three images, there are several possibilities on how to compute the matching cost between multiple corresponding patches. One possible choice is to compute all pairwise similarity measures to the reference patch and average the costs. While this renders the estimation more robust if the actual 3D point is visible in all views, it may lead to deteriorated results in case of occlusions. To deal with this, one can apply the idea of temporal selection [98] and compute all pairwise similarity measures w.r.t. the reference patch, but only consider the lowest pairwise cost as overall cost. Thereby it can be ensured that, as long as the reference patch is visible in at least one additional view, the correct correspondence retains a small cost, even in case it is occluded in the remaining ones (see Figure 6.2 right). In our experiments, we use both approaches, temporal averaging and temporal selection.

Finally, we utilize SIFT descriptors [85, 112, 114] to compute the similarity between two corresponding locations. This choice also renders the matching more robust than operating directly on the intensity values. Regarding the cost function, we follow [85] and apply a robust L^1-loss. The resulting forward and backward structure matching costs C_{t+1} and C_{t-1} are then given by

$$C_{t+1}(\mathbf{x}, z(\mathbf{x})) = |\mathbf{f}_{\text{SIFT}}(\pi(P_{t+1}\tilde{\mathbf{s}}(\mathbf{x}, z)) - \mathbf{f}_{\text{SIFT}}(\pi(P_t\tilde{\mathbf{s}}(\mathbf{x}, z))|_1 \,, \tag{6.5}$$

$$C_{t-1}(\mathbf{x}, z(\mathbf{x})) = |\mathbf{f}_{\text{SIFT}}(\pi(P_{t-1}\tilde{\mathbf{s}}(\mathbf{x}, z)) - \mathbf{f}_{\text{SIFT}}(\pi(P_t\tilde{\mathbf{s}}(\mathbf{x}, z))|_1 \,, \tag{6.6}$$

where $\mathbf{f}_{\text{SIFT}}$ denotes the SIFT-feature and $|\cdot|_1$ is the L^1-norm. The corresponding temporal averaging and temporal selection costs read

$$C_{\text{averaging}}(\mathbf{x}, z) = \frac{1}{2}(C_{t+1}(\mathbf{x}, z) + C_{t-1}(\mathbf{x}, z)) \,, \tag{6.7}$$

$$C_{\text{selection}}(\mathbf{x}, z) = \min(C_{t+1}(\mathbf{x}, z), C_{t-1}(\mathbf{x}, z)) \,. \tag{6.8}$$

6.1.4.4 Outlier Handling

Finally, we extend the standard bi-directional consistency check to our three-view setting. Therefore, we not only estimate the depth values with frame t as the reference view but also with the other two frames $t+1$ and $t-1$ as the reference views. This yields three depth map estimates from the different views, i.e., z_{t-1}, z_t, and z_{t+1}. Then we take the estimated depth value z_t at frame t, project it into the frames $t+1$ and $t-1$, take the estimated depth values z_{t+1} and z_{t-1} there, and project them back to frame t. Finally, we consider the depth values $z_t(\mathbf{x})$ to be valid for which at least one of the two back projections maps to the starting point. This process can be formalized as follows: first, we compute the discrepancies

$$\Delta_{t+1} := \left|\pi\big(P_t\tilde{\mathbf{s}}_{t+1}\big(\underbrace{\pi(P_{t+1}\tilde{\mathbf{s}}(\mathbf{x}, z))}_{=:\mathbf{x}_{t+1}}, z_{t+1}\big)\big) - \mathbf{x}\right| , \tag{6.9}$$

$$\Delta_{t-1} := \left|\pi\big(P_t\tilde{\mathbf{s}}_{t-1}\big(\underbrace{\pi(P_{t-1}\tilde{\mathbf{s}}(\mathbf{x}, z))}_{=:\mathbf{x}_{t-1}}, z_{t-1}\big)\big) - \mathbf{x}\right| , \tag{6.10}$$

and second, we check if the discrepancies fall below a certain threshold τ that allows us to account for minor inaccuracies

$$z_{t,\text{valid}} = \begin{cases} \text{valid}, & \min(\Delta_{t-1}, \Delta_{t+1}) < \tau. \\ \text{not valid}, & \text{otherwise.} \end{cases} \tag{6.11}$$

Finally, for all the valid depth values, we can compute the forward/backward structure matches from $z_t(\mathbf{x})$ via Equation 6.2 and Equation 6.3.

6.1.5 Combining Matches

At this point, we have computed filtered forward and backward structure matches from frame t to frames $t+1$ and $t-1$. For the sake of clarity let us denote these matches by $\mathbf{w}_{\text{st, fw}}$ and $\mathbf{w}_{\text{st, bw}}$. Moreover, as indicated in Figure 6.1 we also computed the corresponding forward and backward optical flow matches between the same frames with a hierarchical PatchMatch approach for unconstrained motion [85]. These optical flow matches underwent a classical bi-directional consistency check to remove outliers (which requires to additionally compute matches from frames $t+1$ and $t-1$ to frame t), let us denote them by $\mathbf{w}_{\text{of, fw}}$ and $\mathbf{w}_{\text{of, bw}}$.

The goal of the combination step is now to fuse these four matches in a way such that rigid parts of the scene can benefit from the structure matches. Thereby one has to keep in mind that optical flow matches may explain rigid motion, while structure matches are typically wrong in the context of independent object motion. To avoid using structural matches at inappropriate locations, we hence propose a conservative approach: We augment the optical flow matches with the matches obtained from the structure matching. This means that we always keep the match of the forward flow if it has passed the outlier filtering. Otherwise, however, we consider augmenting the final matches at this location by the match of the structure matching approach. To decide if we should consider such a structure match, we propose three different approaches ranging from a permissive approach to the point of a restrictive approach.

Permissive Approach The first approach is the most permissive approach. It includes all structure matches $\mathbf{w}_{\text{st, fw}}$ that have passed the outlier filtering at locations where no forward optical flow match $\mathbf{w}_{\text{of, fw}}$ is available, see Figure 6.4.

Restrictive Approach The second approach is more restrictive than the previous one. Instead of including all structure matches, we enforce an additional consistency check. This check allows reducing the probability of blindly including possibly false matches drastically. To realize it, we make use of the backward optical flow match $\mathbf{w}_{\text{of, bw}}$. We only consider the forward structure match $\mathbf{w}_{\text{st, fw}}$, if its backward variant $\mathbf{w}_{\text{st, bw}}$ is consistent with the backward optical flow match $\mathbf{w}_{\text{of, bw}}$. In case the additional consistency check cannot be performed, because the backward optical flow match did not pass the outlier filtering, we do not consider the structure match, see Figure 6.4.

Voting Approach Finally, we propose a voting approach that enforces the additional consistency check as in the restrictive approach but still allows to include structure matches in cases where we cannot perform the additional consistency check. The decision of whether we should include such non-checkable structure matches is conducted for each sequence separately. It uses a voting scheme: All locations, that contain a valid match for the forward, backward and structure match are eligible to vote. If the structure match is consistent with both the forward and

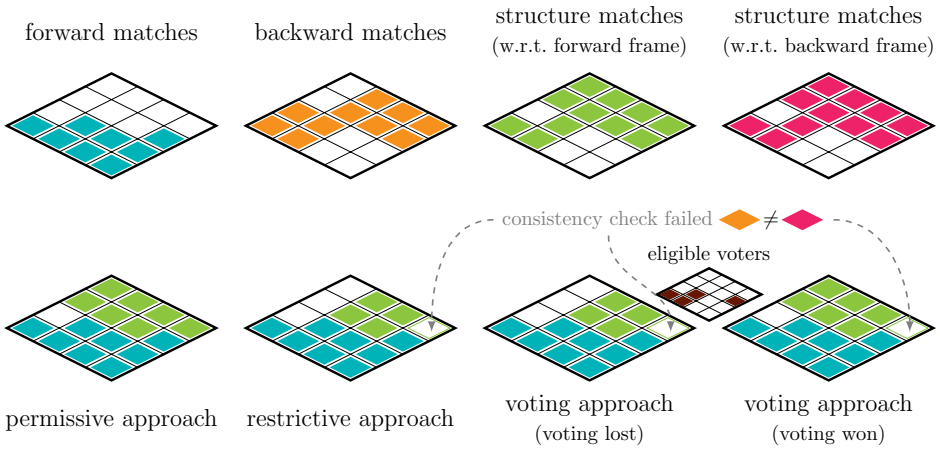


Figure 6.4: Illustration showing the different strategies to combine the computed matches. *Top:* Color coded input matches, where white denotes no match. *Bottom:* Fusion results.

the backward match, we count this as a vote in favor of including non-checkable matches. If the votes surpass a certain threshold (80% in our experiments), we include all non-checkable structure matches, see Figure 6.4. This procedure can be seen as a detection scheme that allows identifying scenes with a large amount of ego-motion. In this case, it might indeed be beneficial to include as many structure matches as possible.

6.1.6 Evaluation Part 1

Evaluation Setup To evaluate our new approach, we used the following components within our pipeline (cf. Figure 6.1): The pose estimation uses the OpenMVG [123] implementation of the incremental SfM approach [122], the forward and backward matching employ the coarse-to-fine PatchMatch (CPM) [85] approach, the structure matching and consistent combination are performed as described in Subsection 6.1.4 and Subsection 6.1.5, respectively, followed by a robust interpolation of the combined correspondences (RIC) using [84]. Finally, the inpainted matches are refined using the order-adaptive illumination-aware refinement method (OIR) as described in Chapter 5. Except for the refinement, where we optimized [14] the three weighting parameters per benchmark using the training data (see Section B.4), we used the default parameters.

Benchmarks To evaluate the performance of our approach, we consider the three most popular benchmarks: the KITTI 2012 [65], the KITTI 2015 [119], and the MPI Sintel [44] benchmark. These benchmarks exhibit an increasing amount of ego-motion induced optical flow. While KITTI 2012 consists of pure ego-motion, KITTI 2015 additionally includes the motion of other traffic participants. Finally, MPI Sintel also contains non-rigid motion from animated characters.

Baseline To measure improvements, we establish a baseline that does not use structure information and only relies on forward optical flow matches (CPM). As Table 6.1 shows, our baseline outperforms most of the related approaches. Only DF+OIR, which we introduced in Chapter 5, performs slightly better, due to the more advanced matches produced by DiscreteFlow [120].

Table 6.1: Results for the training datasets of the KITTI 2012 [65] (all pixels), KITTI 2015 [119] (all pixels) and the MPI Sintel [44] benchmarks (clean render path) in terms of the average endpoint error (AEE) and the percentage of bad pixels (BP, 3px threshold).

Method				KITTI 2012		KITTI 2015		Sintel
name	matching	inpainting	refinement	AEE	BP	AEE	BP	AEE
related approaches (+ baseline)								
CPM-Flow [85]	CPM	EPIC	EPIC	3.00 px	14.58 %	7.78 px	22.86 %	2.00 px
RIC-Flow [84]	CPM	RIC	OpenCV	2.94 px	10.94 %	7.24 px	21.46 %	2.16 px
CPM+OIR [8]	CPM	EPIC	OIR	2.78 px	9.68 %	7.36 px	19.21 %	1.99 px
DF+OIR [8]	DF	EPIC	OIR	**2.34** px	9.29 %	**5.89** px	**18.10** %	**1.91** px
baseline	CPM	RIC	OIR	2.61 px	**8.98** %	6.82 px	18.70 %	1.95 px
only structure matching								
two-frame	CPMz	RIC	OIR	2.25 px	9.47 %	9.15 px	23.02 %	17.09 px
temporal averaging	CPMz	RIC	OIR	**1.25** px	**6.51** %	**7.85** px	**19.11** %	20.68 px
temporal selection	CPMz	RIC	OIR	1.43 px	6.69 %	8.06 px	19.52 %	**15.69** px
only unconstrained matching								
backward flow	CPM	RIC	OIR	6.90 px	43.96 %	11.57 px	44.12 %	4.00 px
forward flow	CPM	RIC	OIR	**2.61** px	**8.98** %	**6.82** px	**18.70** %	**1.95** px
combined fw&bw	CPM	RIC	OIR	4.53 px	18.93 %	9.54 px	27.42 %	2.05 px
combined (temporal selection)								
permissive approach	CPM/CPMz	RIC	OIR	**1.47**px	5.91 %	4.95 px	14.12 %	2.53 px
restrictive approach	CPM/CPMz	RIC	OIR	1.60 px	6.22 %	5.20 px	15.10 %	**1.88** px
voting approach	CPM/CPMz	RIC	OIR	1.48 px	**5.82** %	**4.91** px	**13.95** %	1.90 px
combined (temporal averaging)								
permissive approach	CPM/CPMz	RIC	OIR	**1.30** px	5.71 %	4.21 px	13.72 %	2.92 px
restrictive approach	CPM/CPMz	RIC	OIR	1.59 px	6.17 %	5.04 px	14.97 %	**1.90** px
voting approach	CPM/CPMz	RIC	OIR	**1.30** px	**5.67** %	**4.16** px	**13.61** %	1.92 px
recent literature								
PWC-Net [163]	CVPR '18			4.14 px	–	10.35 px	33.67 %	2.55 px
FlowNet2 [89]	CVPR '17			4.09 px	–	10.06 px	30.37 %	2.02 px
UnFlow [118]	AAAI '18			**3.29** px	–	**8.10** px	23.27 %	–
DCFlow [198]	CVPR '17			–	–	–	15.09 %	–
MR-Flow [196]	CVPR '17			–	–	–	14.09 %	**1.83** px
Mirror Flow [87]	ICCV '17			–	–	–	**9.98** %	–
learning approaches (fine tuned)								
PWC-Net-ft [163]	CVPR '18			(1.45 px)	–	(2.16 %)	(9.80 px)	(1.70 px)
FlowNet2-ft [89]	CVPR '17			(1.28 px)	–	(2.30 %)	(8.61 px)	(1.45 px)
UnFlow-ft [118]	AAAI '18			(**1.14** px)	–	(**1.86** %)	(**7.40** px)	–

Structure Matching Next, we investigate the performance of our novel structure matching approach on its own. Therefore, we replace the matching approach (CPM) in our baseline with three variants of our structure matching approach (CPMz): a two-frame variant, a three-frame variant with temporal averaging and a three-frame variant with the temporal selection. As the results in Table 6.1 show, structure matching significantly outperforms the baseline in pure ego-motion scenes (e.g., KITTI 2012), while it naturally has problems in scenes with independent motion (e.g., MPI Sintel). Moreover, they show that the use of multiple frames pays off. While for the KITTI benchmarks the robustness of temporal averaging is more beneficial than the occlusion handling of temporal selection, the opposite holds for the MPI Sintel benchmark. This behavior, in turn,

might be attributed to the fact that MPI Sintel contains a more considerable amount of occlusions. Since both strategies have their advantages, we consider both variants for our further evaluation.

Unconstrained Matching Apart from the baseline we also evaluated two additional variants solely based on unconstrained matching: a variant only using backward matches and a variant that augments the forward matches with backward matches. In both cases we assume a constant motion model, i.e., $\mathbf{w}_{\text{of, fw}} = -\mathbf{w}_{\text{of, bw}}$. The results for the backward flow in Table 6.1 show that such a simple model does not allow to leverage useful information to predict the forward flow. Even the augmented variant does not improve compared to the baseline. Visual exemplary results for the three benchmarks are given in Figure 6.5 and Figure 6.6.

Combined Approach Let us now turn towards the evaluation of our combined approach. In this context, we compare the impact of the different combination strategies. As one can see in Table 6.1, the permissive approach is not an option. While it works well for dominating ego-motion, it includes too many false structure matches in case of independent object motion, see also Figure 6.6. In contrast, the restrictive approach prevents the inclusion of false structure matches, but cannot make use of the full potential of such structure matches in scenes with dominating ego-motion. Nevertheless, it already outperforms the baseline significantly and gives the best results for MPI Sintel. Finally, the voting approach combines the advantages of both schemes. It yields the best results for KITTI 2012 and 2015 with improvements up to 50% compared to the baseline, while still offering an improvement w.r.t. MPI Sintel. The examples in Figure 6.5 and Figure 6.6 also confirm this observation, where we compare the three combination strategies visually. They not only show the usefulness of including structure matches in occluded areas, but also the importance of filtering false structure matches in general. Moreover, they show that in contrast to using backward matches in occluded areas, structure matches offer an appropriate motion model to leverage additional temporal information for the rigid background parts of the scene.

Comparison to the Literature Next, we compare our method to other approaches from the literature. To this end, we consider both the training and the test data; see Table 6.1, Table 6.2 and Table 6.3, respectively. Regarding the training data, our method generally yields better results than recent learning approaches without fine-tuning (PWC-Net [163], FlowNet2 [89], UnFlow [118]). Moreover, it also outperforms DCFlow [198] and MR-Flow [196] on the KITTI 2015 benchmark. Only MirrorFlow [87] (KITTI 2015) and MR-Flow (MPI Sintel) provide better results than our approach. This excellent performance is also confirmed by the results of our method for the test data as well, for which we evaluated the approaches that had performed best on the training data. Here, on KITTI 2012, our method performs favorably (all pixels) even compared to methods based on pure ego-motion (SPS-Fl, PCBP-Flow, and MotionSLIC) and semantic information (SDF). Moreover, it also outperforms recent approaches with an explicit SfM background estimation (MR-Flow) on KITTI 2015. Finally, ranking second and sixth at the time of submission (Mar. 2018) our method also yields an excellent performance on the clean and the final render path of the MPI Sintel benchmark, respectively. These results demonstrate that our method not only works well in the context of pure ego-motion but can also handle a significant amount of independent object motion.

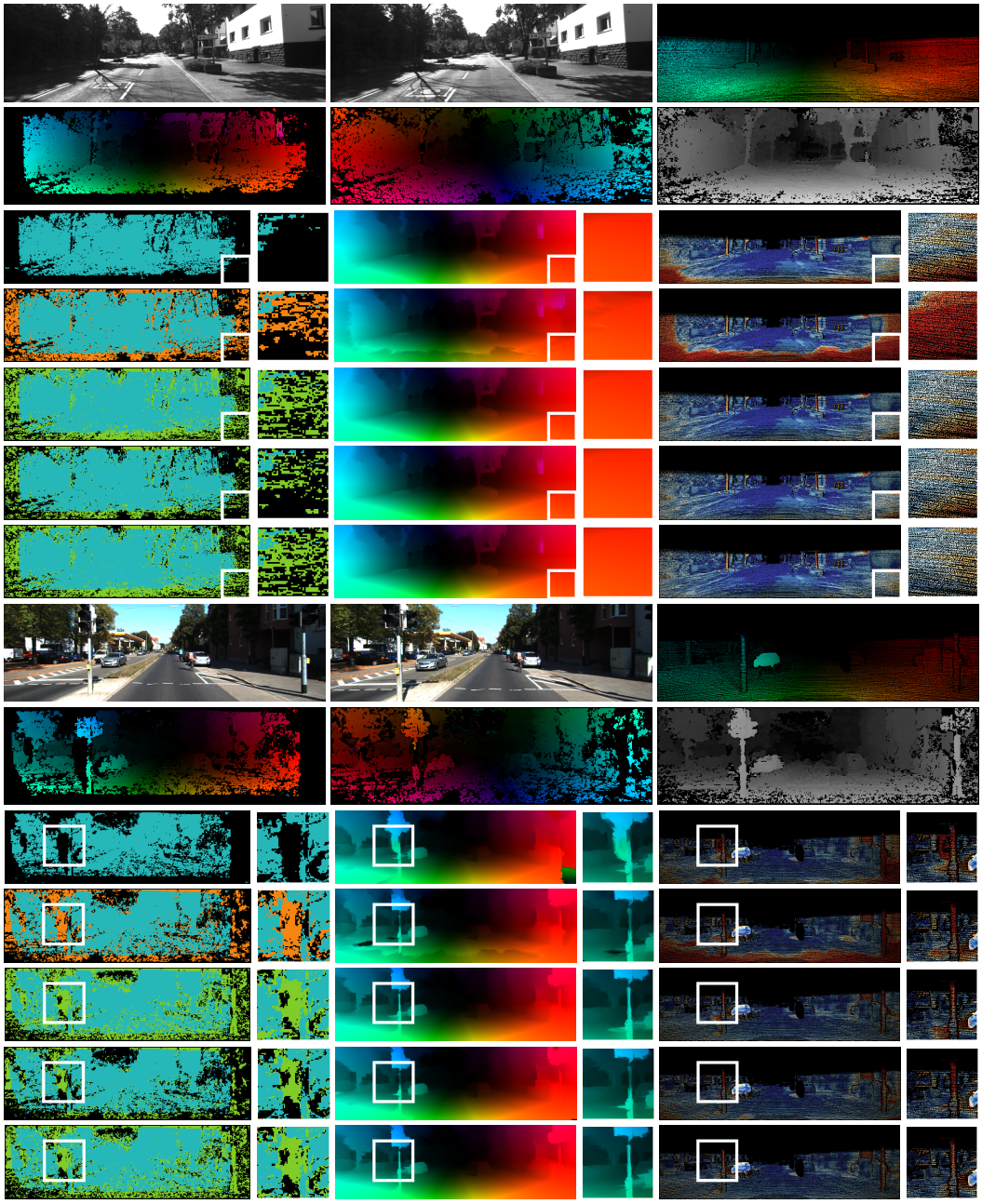


Figure 6.5: Example for the KITTI 2012 [65] and KITTI 2015 benchmark [119] (seq. #0 and seq. #186). *First row:* Reference frame, subsequent frame, ground truth. *Second row:* Forward, backward, and structure matches (depth visualization). *Third to seventh row. From left to right:* Used matches (color-coding see Figure 6.4), final result, error visualization. *From top to bottom:* Baseline, combined forward and backward matches (constant motion model), permissive approach (rigid motion model), restrictive approach (rigid motion model), voting approach (rigid motion model). *Remaining rows:* Same as for the previous rows.

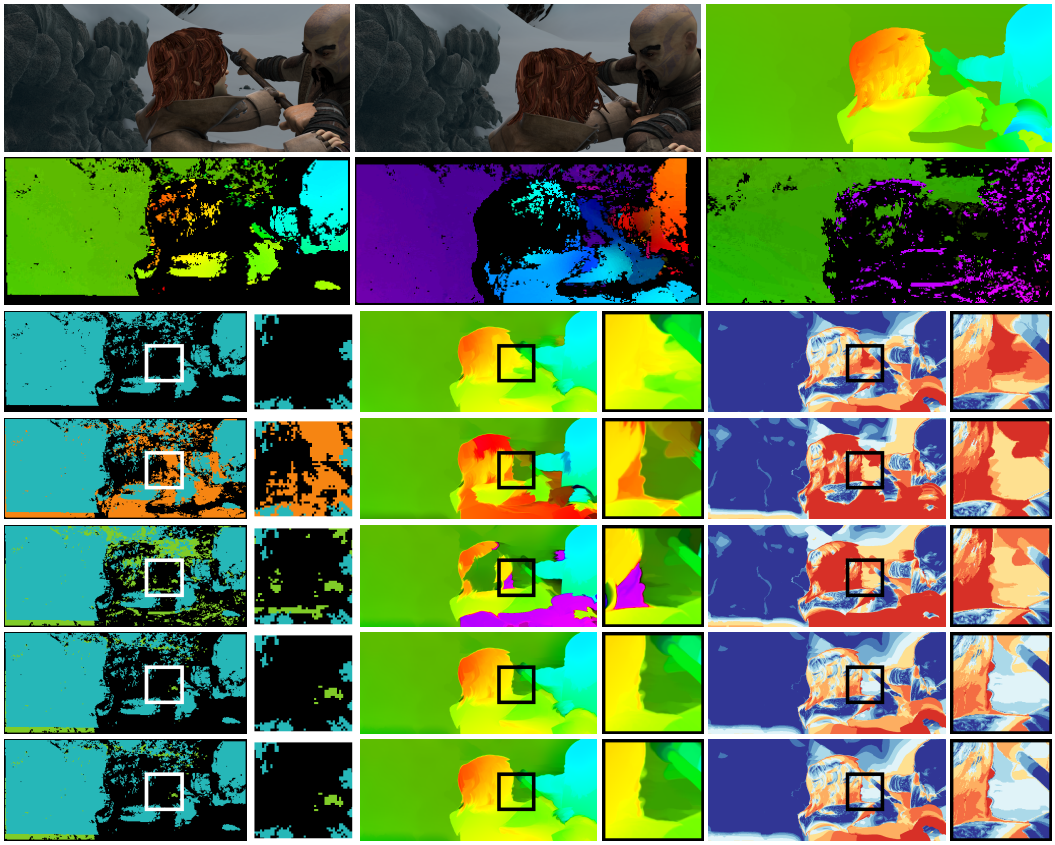


Figure 6.6: Example for the MPI Sintel benchmark [44] (ambush5 #44). *First row:* Reference frame, subsequent frame, ground truth. *Second row:* Forward, backward, and structure matches (forward match visualization). *Following rows. From left to right:* Used matches (color-coding see Figure 6.4), final result, bad pixel visualization. *From top to bottom:* Baseline, combined forward and backward matches (constant motion model), permissive approach (rigid motion model), restrictive approach (rigid motion model), voting approach (rigid motion model).

6.1.7 Evaluation Part 2

In addition to the experiments in the previous section, we next conduct several experiments regarding individual benefits of parameter settings, design choices, and components.

Fixed Parameter Set In the following experiment, we investigate how the results change when not optimizing the refinement parameters individually for each benchmark. To this end, we considered the voting approach with temporal averaging and conducted an experiment on the training data with all parameters fixed. As Table 6.4 shows the results hardly deteriorate when using a single parameter set for all benchmarks.

Pose Estimation So far we used all available frames of the image sequence for the pose estimation jointly in our experiments. To investigate the impact of this choice on the final result, we consider an additional variant: A minimal three-frame variant that uses only the three frames

Table 6.2: Top 10 non-anonymous optical flow methods and methods presented in this thesis on the test data of the KITTI 2012/2015 [65, 119] at the time of submission (Mar. 2018), excluding scene flow methods. The presented approach is highlighted in red. The methods presented in Chapter 5 (DF+OIR), Section 4.1 (SODA-Flow) and Section 4.2 (OAR-Flow) are highlighted in blue.

KITTI 2012	Out-Noc	Out-All	Avg-Noc	Avg-All	KITTI 2015	Fl-bg	Fl-fg	Fl-all
SPS-Fl[1] [200]	3.38 %	10.06 %	0.9 px	2.9 px	PWC-Net [163]	9.66 %	9.31 %	9.60 %
PCBP-Flow[1] [201]	3.64 %	8.28 %	0.9 px	2.2 px	MirrorFlow [87]	8.93 %	17.07 %	10.29 %
SDF[2] [19]	3.80 %	7.69 %	1.0 px	2.3 px	SDF[2] [19]	8.61 %	23.01 %	11.01 %
MotionSLIC[1] [201]	3.91 %	10.56 %	0.9 px	2.7 px	UnFlow [118]	10.15 %	15.93 %	11.11 %
our approach	**4.02** %	**6.15** %	**1.0** px	**1.5** px	CNNF+PMBP [212]	10.08 %	18.56 %	11.49 %
PWC-Net [163]	4.22 %	8.10 %	0.9 px	1.7 px	our approach	**9.66** %	**22.73** %	**11.83** %
UnFlow [118]	4.28 %	8.42 %	0.9 px	1.7 px	MR-Flow[2] [196]	10.13 %	22.51 %	12.19 %
MirrorFlow [87]	4.38 %	8.20 %	1.2 px	2.6 px	DCFlow [198]	13.10 %	23.70 %	14.86 %
ImpPB+SPCI [154]	4.65 %	13.47 %	1.1 px	2.9 px	SOF[2] [156]	14.63 %	22.83 %	15.99 %
CNNF+PMBP [212]	4.70 %	14.87 %	1.1 px	3.3 px	JFS[2] [86]	15.90 %	19.31 %	16.47 %
DF+OIR [8]	5.17 %	10.43 %	1.1 px	2.9 px	DF+OIR [8]	15.11 %	23.45 %	16.50 %
SODA-Flow [10]	5.57 %	10.71 %	1.3 px	2.8 px	SODA-Flow [10]	20.01 %	29.14 %	21.53 %
OAR-Flow [9]	5.69 %	10.72 %	1.4 px	2.8 px	OAR-Flow [9]	20.62 %	27.67 %	21.79 %

[1] uses epipolar geometry as a hard constraint, only applicable to pure ego-motion
[2] exploits semantic information

Table 6.3: Top 10 non-anonymous optical flow methods and methods presented in this thesis on the test data of the MPI Sintel benchmark [44] at the time of submission (Mar. 2018) in terms of the average endpoint error. The presented approach is highlighted in red. The methods presented in Chapter 5 (DF+OIR) and Section 4.2 (OAR-Flow) are highlighted in blue.

MPI Sintel clean	all	matched	unmatched	MPI Sintel final	all	matched	unmatched
MR-Flow[2] [196]	2.527 px	0.954 px	15.365 px	PWC-Net [163]	5.042 px	2.445 px	26.221 px
our approach	**2.910** px	**1.016** px	**18.357** px	DCFlow [198]	5.119 px	2.283 px	28.228 px
FlowFields+ [21]	3.102 px	0.820 px	21.718 px	FlowFieldsCNN [22]	5.363 px	2.303 px	30.313 px
CPM2 [111]	3.253 px	0.980 px	21.812 px	MR-Flow[2] [196]	5.376 px	2.818 px	26.235 px
MirrorFlow [87]	3.316 px	1.338 px	19.470 px	S2F-IF [203]	5.417 px	2.549 px	28.795 px
DF+OIR [8]	3.331 px	0.942 px	22.817 px	our approach	**5.466** px	**2.683** px	**28.147** px
S2F-IF [203]	3.500 px	0.988 px	23.986 px	InterpoNet_ff [219]	5.535 px	2.372 px	31.296 px
SPM-BPv2 [110]	3.515 px	1.020 px	23.865 px	RicFlow [84]	5.620 px	2.765 px	28.907 px
DCFlow [198]	3.537 px	1.103 px	23.394 px	InterpoNet_cpm [219]	5.627 px	2.594 px	30.344 px
RicFlow [84]	3.550 px	1.264 px	22.220 px	ProbFlowFields [181]	5.696 px	2.545 px	31.371 px
–	–	–	–	DF+OIR [8]	5.862 px	2.864 px	**30.303** px
OAR-Flow [9]	6.227 px	2.760 px	34.455 px	OAR-Flow [9]	8.179 px	4.578 px	37.525 px

[1] uses epipolar geometry as a hard constraint, only applicable to pure ego-motion
[2] exploits semantic information

Table 6.4: Impact of refinement parameter optimization (temporal averaging setting).

method		KITTI 2012		KITTI 2015		Sintel
name	parameters	AEE	BP	AEE	BP	AEE
voting approach	individually optimized	**1.30** px	**5.67** %	**4.16** px	**13.61** %	**1.92** px
voting approach	single parameter set	1.31 px	5.70 %	**4.16** px	13.70 %	1.93 px

of the structure matching step and computes the three relative poses independently from all the other frames in a sliding window fashion. Please note in this context that the advantages and drawbacks of the two choices are quite complimentary. While the full-sequence approach may offer benefits for smooth sequences with small motion, the three-frame approach is less likely to propagate problems with estimating single poses to all frames. As one can see from Table 6.5, the performance even slightly improves in case of the permissive and the restrictive approach when considering only three frames. In case of the voting approach, such a gain is only observed for the KITTI benchmarks.

Variational Refinement Finally, we investigate the impact of the variational refinement on the results of the baseline and of our SfM-aware PatchMatch approach. As one can see from Table 6.6, the variational refinement yields a consistent improvement of the results for all benchmarks and all methods.

Runtime On average, the runtime of our pipeline excluding the pose estimation is 32s for one frame of size 1024×436 (MPI Sintel) using three cores on an Intel® Core™ i7-7820X CPU @ 3.6GHz, which splits into 5.5s matching (incl. outlier filtering), <0.1s combination, 1.5s inpainting, and 25s refinement. The pose estimation run on the entire image sequence takes 83s for a sequence with 50 frames. The three-frame-variant needs 6s per frame which sums up to 300s for the entire sequence. This demonstrates that a three-frame approach is a valid option regarding the estimation quality when a sequential computation of the frames is required. However, it also shows that from a computational viewpoint, the full-sequence approach is the better alternative.

Table 6.5: Impact of using a different number of frames for the pose estimation on the results for the training datasets of the KITTI 2012 [65] (all pixels), KITTI 2015 [119] (all pixels) and the MPI Sintel [44] benchmarks (clean render path) in terms of the average endpoint error (AEE) and the percentage of bad pixels (BP, 3px threshold).

method		KITTI 2012		KITTI 2015		Sintel
name	# frames used for pose estimation	AEE	BP	AEE	BP	AEE
baseline	–	2.61 px	8.98	6.82	18.70	1.95
our approach (temporal selection)						
permissive approach	3 frames	**1.42** px	**5.58** %	4.81 px	13.79 %	2.24 px
permissive approach	complete sequence	1.47 px	5.91 %	4.95 px	14.12 %	2.53 px
restrictive approach	3 frames	1.59 px	6.04 %	5.10 px	15.00 %	**1.87** px
restrictive approach	complete sequence	1.60 px	6.22 %	5.20 px	15.10 %	1.88 px
voting approach	3 frames	**1.42** px	5.60 %	**4.78** px	**13.78** %	1.92 px
voting approach	complete sequence	1.48 px	5.82 %	4.91 px	13.95 %	1.90 px
our approach (temporal averaging)						
permissive approach	3 frames	**1.28** px	**5.47** %	4.09 px	13.45 %	2.43 px
permissive approach	complete sequence	1.30 px	5.71 %	4.21 px	13.72 %	2.92 px
restrictive approach	3 frames	1.57 px	6.00 %	5.05 px	14.86 %	**1.88** px
restrictive approach	complete sequence	1.59 px	6.17 %	5.04 px	14.97 %	1.90 px
voting approach	3 frames	1.30 px	5.52 %	**4.07** px	**13.41** %	1.98 px
voting approach	complete sequence	1.30 px	5.67 %	4.16 px	13.61 %	1.92 px

Table 6.6: Impact of the variational refinement [8] on the results for the training datasets of the KITTI 2012 [65] (all pixels), KITTI 2015 [119] (all pixels) and the MPI Sintel [44] benchmarks (clean render path) in terms of the average endpoint error (AEE) and the percentage of bad pixels (BP, 3px threshold).

method name	matching	inpainting	refinement	KITTI 2012 AEE	KITTI 2012 BP	KITTI 2015 AEE	KITTI 2015 BP	Sintel AEE
baseline	CPM	RIC	–	2.89 px	9.73 %	7.20 px	19.96 %	2.25 px
baseline	CPM	RIC	OIR	**2.61** px	**8.98** %	**6.82** px	**18.70** %	**1.95** px
combined (temporal selection)								
permissive approach	CPM/CPMz	RIC	–	1.71 px	6.20 %	5.29 px	14.76 %	2.82 px
permissive approach	CPM/CPMz	RIC	OIR	**1.47** px	5.91 %	4.95 px	14.12 %	2.53 px
restrictive approach	CPM/CPMz	RIC	–	1.84 px	6.56 %	5.53 px	15.91 %	2.11 px
restrictive approach	CPM/CPMz	RIC	OIR	1.60 px	6.22 %	5.20 px	15.10 %	**1.88** px
voting approach	CPM/CPMz	RIC	–	1.72 px	6.13 %	5.25 px	14.64 %	2.13 px
voting approach	CPM/CPMz	RIC	OIR	1.48 px	**5.82** %	**4.91** px	**13.95** %	1.90 px
combined (temporal averaging)								
permissive approach	CPM/CPMz	RIC	–	1.55 px	6.06 %	4.58 px	14.58 %	3.25 px
permissive approach	CPM/CPMz	RIC	OIR	**1.30** px	5.71 %	4.21 px	13.72 %	2.92 px
restrictive approach	CPM/CPMz	RIC	–	1.84 px	6.53 %	5.39 px	15.82 %	2.13 px
restrictive approach	CPM/CPMz	RIC	OIR	1.59 px	6.17 %	5.04 px	14.97 %	**1.90** px
voting approach	CPM/CPMz	RIC	–	1.55 px	5.99 %	4.53 px	14.46 %	2.14 px
voting approach	CPM/CPMz	RIC	OIR	**1.30** px	**5.67** %	**4.16** px	**13.61** %	1.92 px

Figure 6.7: Example sequences of the KITTI 2012/2015 [65, 119] and MPI Sintel benchmarks [44] that do not allow to estimate a pose using the applied pose estimation [122, 123].

6.1.8 Limitations

The main limitation of our method is its dependency on a reliable pose estimation. Hereby, one can distinguish two scenarios. In case of a (partially) non-valid pose, the integration of structure information typically does not pose a problem, since possibly false matches are mostly eliminated by the additional consistency check that forms the basis of the restrictive approach and the voting approach; see Figure 6.6. Only if the pose estimation fails completely, i.e., the underlying algorithm does not provide any pose, we cannot obtain any structure matches. In this case, however, we can still rely on the forward matches which, in turn, comes down to using the baseline approach. In

Figure 6.7 we depict example sequences where the pose estimation fails, i.e., a sequence without ego-motion (camera is stationary), a scene with purely rotational motion (no camera translation) and a scene with a large-dominant non-rigid foreground object. While it might be possible to recover the pose in those scenarios using specifically tailored algorithms, we refrained from this option, since the focus of our method lies on the integration of structure information rather than on estimating the pose itself.

6.1.9 Conclusion

In this section, we proposed a multi-frame method by integrating structure information into feature matching approaches for computing the optical flow. To this end, we developed a hierarchical depth-parametrized three-frame SfM/stereo PatchMatch approach with a temporal selection and preceding pose estimation. By adaptively combining the resulting matches with those of a recent PatchMatch approach for general motion estimation, we obtained a novel SfM-aware method that benefits from a global rigidity prior, while still being able to estimate independently moving objects. Experiments not only showed excellent results on all major benchmarks (KITTI 2012, KITTI 2015, and MPI Sintel), they also demonstrated consistent improvements over a baseline without structure information. Since our approach addresses the first step of common pipeline based approaches, it also offers another advantage: incorporating our matches as initialization into other pipeline approaches allows them to easily benefit from them.

6.2 Learned Motion Model

After introducing an approach based on a rigid motion model, we also propose an approach that allows to adapt the underlying motion model by learning an appropriate model *during the estimation*. This new strategy allows us to overcome several limitations of the rigid motion model and thus exploit multi-frame information even for independently moving objects, non-ego motion scenes, and non-rigid motion scenarios. To realize our novel idea we, once more, built upon the pipeline approach as introduced in the previous chapter (Chapter 5). But before we dive into the explanation of our novel method, we first take a look at related work.

6.2.1 Related Work

Multi-Frame Approaches To improve the quality and the robustness of the estimation, multi-frame strategies typically build upon some motion model that describes how the movement is expected to change over time. In this context, recent approaches go far beyond a simple constant velocity model [91, 92, 99, 193] by using constraints based on constant acceleration [28, 160, 179], parametrized trajectories [64, 144] or a moving camera [6, 196]. Moreover, to avoid a significant deterioration of the results in case the model turns out to be inappropriate, they typically allow deviations from the model either by formulating it as a soft constraint [28, 64, 160, 179] or by restricting the estimation to locations where the assumed model is most likely to hold [179, 196]. Compared to most of the methods mentioned above, our method differs in two ways: On the one hand, our approach does not use hand-crafted or geometric/rigid motion models but learns spatially varying mappings from the backward to the forward flow. On the other hand, our approach

uses the learned motion models as a hard constraint, i.e., without any filtering and at all locations where the backward flow provides additional information, e.g., at occlusions.

Regarding the learning of motion models, two other interesting approaches have been proposed by Ricco and Tomasi [144] and Garg et al. [64] that learn temporal basis functions for long term trajectories via PCA from pre-computed tracks and flow fields, respectively. However, in contrast to these approaches that focus on a robust long term motion representation to perform dense tracking and non-rigid video registration, respectively, our new method aims at a short-term optical flow setting and provides state-of-the-art results for standard optical flow benchmarks.

Since the time of publication [2], others have followed our idea and considered such a multi-frame setting in the context of end-to-end learning based optical flow approaches. However, instead of modeling explicit motion models these approaches typically simply feed an additional preceding flow field estimate to the network and leave it to the network how to relate and include the additional information. For example, Ren et al. [142] train a convolutional neural network (CNN) based fusion network that allows combining two temporal correlated flow fields into a fused flow field. Similarly, Neoral et al. [126] feed a preceding flow field estimate directly into intermediate layers of the network as well as additionally estimated occlusion maps.

Learning Approaches Regarding learning approaches for optical flow estimation, one can distinguish two types of methods: entirely learning-based methods and partially learning-based methods. Entire learning-based methods aim at deriving an end-to-end relation between the input images and the corresponding flow field, typically via one or multiple stacked CNNs [57, 89, 139, 142, 163]. While the overall learning process is quite time-consuming and typically requires a large amount of training data, the learned models allow computing high-quality flow estimates in real-time [89, 163]. Recently, also unsupervised learning approaches have been considered to tackle the lack of realistic training data; see e.g., [17, 118, 141, 206, 215]. They either replace the ground truth by a proxy ground truth computed with modern optical flow methods [215] or they propose a loss function that does not depend on the ground truth, i.e., by using an image-based registration error [17, 92, 118, 141, 206, 215] or some smoothness constraint on the solution [92, 118, 141, 206]. Partially learning-based approaches, on the other hand, are hybrid methods: They seek to combine the advantages of two worlds. While relying on a transparent global energy minimization framework, they make use of machine learning techniques to replace some difficult task during the modeling or the estimation. Such tasks include descriptor learning [22, 61, 162, 198], instance level segmentation [19], rigidity estimation [196], and semantic scene segmentation [156]. Although our approach is partially-learning-based, since it embeds a CNN into a traditional optical flow pipeline [143], it is entirely different from all aforementioned learning-based approaches. Not only that the learning step solves a different problem, i.e., it predicts a forward flow from a backward flow; also the training itself is completely different. It uses an unsupervised/self-supervised online approach that relies on initial flow estimates to train the network individually for each frame of the sequence at runtime instead of training the network once for an entire task based on previously collected set of training data. In this respect, our approach is also intrinsically different from the unsupervised methods listed above. Also those methods do not consider the actual frames when training the underlying network prior to the estimation.

6.2.2 Contributions

As mentioned before, the second multi-frame approach we propose in this chapter relies on a learning-based strategy. Instead of assuming a moving camera with certain rigidity constraints, the proposed method learns suitable motion models based on a CNN. In this context, our contributions are fourfold: (i) In contrast to other approaches that train a network before the estimation, our approach learns the models online, i.e., during the estimation. (ii) Moreover, instead of relying on potentially unsuitable data sets with ground truth, our models are trained using initial flow estimates of the actual sequence. Such an unsupervised/self-supervised training offers the advantage that we can learn appropriate models for each sequence. (iii) Thirdly, our approach not only learns one model per sequence but one model for each frame of every sequence. This per-frame learning results in a high degree of adaptability when it comes to a change of the scene content. (iv) Finally, the learned models are spatially variant, i.e., location dependent. This, in turn, addresses the problem of independently moving objects.

Having learned such dedicated motion models eventually enables us to predict the forward flow from the backward flow. Thus it becomes possible to improve the estimation at locations where the forward flow is not available, e.g., in occluded regions. Experiments make the benefits of our novel method explicit. They show not only consistent improvements compared to a baseline approach without prediction but also excellent results for all major benchmarks in general.

6.2.3 Our Approach

Let us start by giving a brief overview of the proposed method, which we illustrate in Figure 6.8. Please note that, as in case of the previous rigid motion model approach, our method considers image triplets, i.e., the frames at times $t-1$, t, and $t+1$. Compared to classical two-frame approaches, this additional frame allows to compute the optical flow from the reference frame t not only to the subsequent frame $t+1$ (forward flow) but also to the previous frame $t-1$ (backward flow), see Figure 6.9. After we have estimated both flow fields with a conventional optical flow approach, we perform outlier filtering via a bi-directional consistency check. While this requires the additional computation of flow fields using the reversed frame order, it allows us to identify possibly occluded image regions. Based on locations where both the forward and the backward flow are available

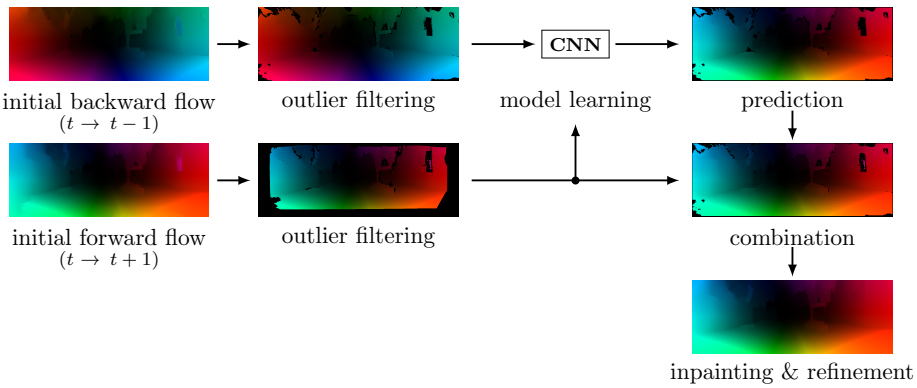


Figure 6.8: Schematic overview over our proposed approach.

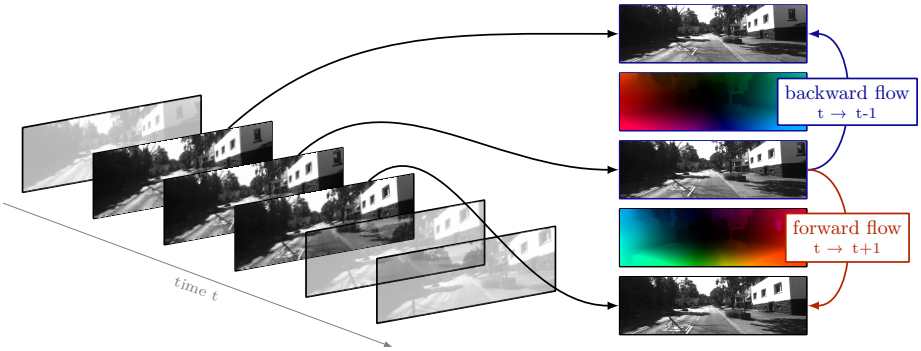


Figure 6.9: Sketch illustrating the meaning of forward and backward flow.

after filtering, we then learn a model that allows predicting the forward flow from the backward flow. To this end, we train a CNN such that it performs a regression from small backward flow patches to forward flow vectors. Using the trained network, we then predict a new forward flow field from the filtered backward flow. This prediction provides additional information at those locations where only the backward flow is given, e.g., at occlusions. Finally, the predicted and the initial forward flow field are combined such that predictions are used if no initial forward flow is available. As the last step, we inpaint the combined flow field to obtain dense results and refine it to improve its accuracy further. Let us now detail on the different steps of the pipeline.

6.2.3.1 Initial Flow Estimation / Baseline

In a first step, we compute the initial forward and backward flow fields, i.e., the flow fields from the reference frame t to subsequent frame $t + 1$ and from reference frame t to previous frame $t - 1$, respectively. To this end, we consider once more the baseline approach introduced in the previous Section 6.1, which builds upon the optical flow pipeline as described in Chapter 5. In particular, we employ the Coarse-to-fine PatchMatch approach (CPM) of Hu et al. [85] for the matching, the robust interpolation technique (RIC) of Hu et al. [84] for the inpainting of the matches and our order-adaptive illumination-aware refinement (OIR) scheme introduced in Chapter 5 for the final refinement step. Thereby the outlier filtering applied to the initial matches is realized in terms of a bi-directional consistency check. Please note that this check requires to compute matches in the reverse direction as well, i.e., from frame $t + 1$ to frame t and from frame $t - 1$ to frame t, respectively. Moreover, note that the described pipeline is only used to compute the initial flow fields and hence constitutes only one step of the entire optical flow approach, see Figure 6.8.

6.2.3.2 Outlier Filtering

After we have computed the initial forward and backward flow fields with our baseline approach, we apply another outlier filtering step. Analogously to the baseline itself, we again apply the bi-directional consistency check for the outlier filtering. This time, however, not based on the initial matches, but rather based on the dense initial flow fields. Therefore, we also need to compute flow fields in the reverse direction, i.e., from frame $t + 1$ to frame t and from frame $t - 1$ to frame t, respectively. To this end, we just run our baseline approach on the reverse image sequence. Finally,

only those flow vectors are considered valid in the forward and backward flow field which are consistent with the corresponding vectors in the reverse direction. This check allows eliminating many outliers, in particular in occluded regions.

6.2.3.3 Learning a Motion Model

Having the filtered forward and backward flow fields at hand, let us now discuss how the underlying motion model is learned. The goal of this step is to derive the relation between the backward flow and the forward flow which enables us to use the backward flow for predicting the forward flow at locations where the forward flow is not available. Since motion patterns typically vary across different scenes and frames, we do not use a network that we must train in advance on a vast data set with ground truth data [22, 89, 163, 198], but we apply an unsupervised/self-supervised learning approach that trains a CNN during the optical flow estimation – and that individually for each frame of the sequence. As shown in the work of Galliani et al. [63] in the context of predicting surface normals for multi-view stereo, such unsupervised/self-supervised learning techniques can be highly beneficial to densify initially sparse results.

Training Data Extraction The training data required for the learning process is extracted from the initially computed flow fields after outlier filtering. Thereby, all locations where both the forward and the backward flow surpassed the outlier filtering serve as potential training samples. To obtain a reasonably sized and reasonably diverse training set, we sample these potential samples equidistantly using a grid spacing of 10 pixels. Thereby, the input of each training sample consists of stacked 7×7 patches composed of (i) the backward flow components u_{bw} and v_{bw}, (ii) a validity flag $\{0, 1\}$ indicating if the location surpassed the outlier filtering step and (iii) the x- and y-component of the pixel location within the image domain (normalized to $[-1, 1] \times [-1, 1]$). The stacked forward flow components u_{fw} and v_{fw} give the corresponding output. We illustrate the whole process in Figure 6.10 (left). Please note that the training data is extracted automatically per image triplet during the estimation and does not rely on any ground truth information nor manually labeled training data.

CNN-Based Regression With the extracted training samples we now train a motion model in terms of a CNN which allows us to predict the forward flow solely based on the backward flow. The input of the network consists of stacked 7×7 patches including information on the backward flow, the validity and the location as described in the previous paragraph. The output of

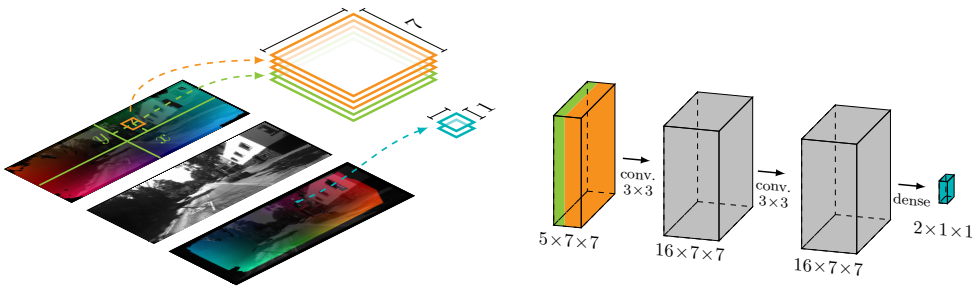


Figure 6.10: *Left:* Training sample extraction. *Right:* Regression network architecture.

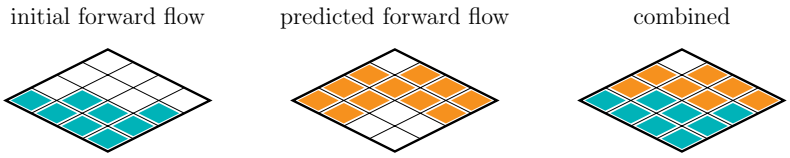


Figure 6.11: Illustration showing the combination step.

the network is the predicted forward flow for the center location of the input patch. By considering not only the backward flows in the input patch but also the corresponding image coordinates, the network is enabled to learn a location-dependent model. This aspect is particularly important, since motion patterns may locally vary due to independently moving objects, non-rigid deformations as well as perspective effects.

Let us now detail on to the architecture and the training process of our regression network. As loss function, we minimize the absolute difference of the predicted flow vector and the actual forward flow vector. Thereby, we keep the network architecture simple, since it has to be trained online for each frame of the sequence: it consists of 2 convolutional layers each with 16 kernels of window size 3×3 and a fully connected layer with a 2-vector output, which represents the desired predicted forward flow vector, as illustrated in Figure 6.10 (right). As non-linearities we employed ReLUs [125]. The network is implemented in the TensorFlow framework [15] and trained using the ADAM optimizer [100] with an exponential learning rate decay. The initial learning rate is set to 0.01 and decays every 200 steps with a base of 0.8. Using the described network and learning scheme 4000 steps were sufficient to train the network.

6.2.3.4 Combination and Final Estimation

After learning the motion model in terms of a CNN, we can use it to predict a new forward flow based on the filtered backward flow. The predicted flow vectors can then be employed to augment the filtered initial forward flow at those locations where no flow vectors are present, as illustrated in Figure 6.11. Since the combined flow field is not dense – at some locations neither forward nor backward flow vectors are available – we finally perform inpainting and refinement with the same techniques as in our baseline; i.e, we use RIC [84] and OIR (Chapter 5, [8]).

6.2.4 Evaluation

To investigate the benefit of our new optical flow approach, which we named ProFlow (*pr*edict *o*ptical *flow*), we consider as before the training data sets as well as the test data sets of the three most popular optical flow benchmarks: the KITTI 2012 benchmark [65], the KITTI 2015 benchmark [119] and the MPI Sintel benchmark [44].

Parameter Setting Regarding the parameters of the used approaches (CPM, RIC, OIR), we used the default parameters as provided by the authors [84, 85, 8]. Consequently, the same set of parameter is used for all benchmarks in case of the matching (CPM) and inpainting (RIC), only in case of the variational refinement (OIR) a different set of parameters is used per benchmark. An exception is given for the results of the Robust Vision Challenge. Here, also a single set of parameters is used for the variational refinement (OIR), see Section B.4.

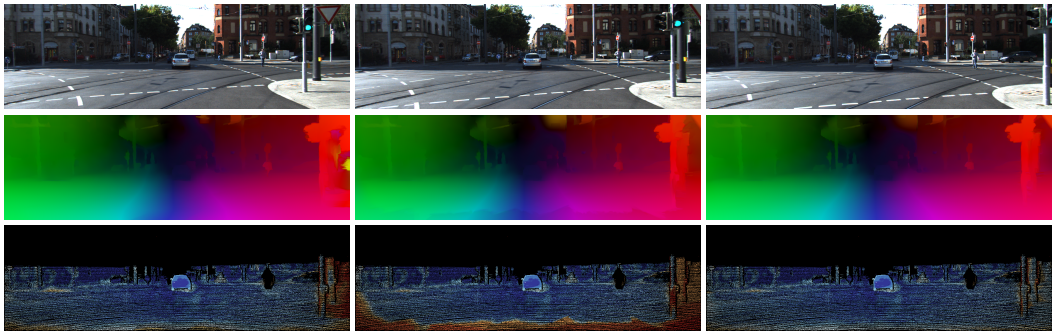


Figure 6.12: Example for the KITTI 2015 benchmark [119] (seq. #149). *First row:* Previous, reference and subsequent frame. *Second and third row:* Estimated flow field, bad pixel visualization. *From left to right:* Baseline, constant motion model and our approach.

Learned vs. Constant Model In our first experiment, we compare our learned motion model with the constant motion model that is frequently used in the literature; see e.g., [91, 92, 99, 193]. This model assumes the forward flow $\mathbf{w}_{\text{fw}}$ and the backward flow $\mathbf{w}_{\text{bw}}$ relate via $\mathbf{w}_{\text{fw}} = -\mathbf{w}_{\text{bw}}$. As already mentioned, this model can be a reasonable approximation in case of slowly moving objects [91], but it typically does not hold for fast or complex motion scenarios [179, 193]. Furthermore, please note that due to the projection involved in the optical flow, such a constant motion model does not represent an actual constant 3D motion unless the motion is parallel to the image plane. For our comparison, we computed the results for the training data sets of all three benchmarks using our approach as well as a modified version, where we omitted the model learning part and directly applied the constant motion model for the prediction. In Table 6.7 (full model) we listed the outcome of both approaches. As one can see, using the constant model for predicting the optical flow does not work well for the challenging benchmarks and even leads to a strong deterioration of the results compared to the baseline. Our approach, in contrast, learns an appropriate motion model and consistently achieves improvements ranging from 8 to 27 percent. The visual comparison in Figure 6.12, Figure 6.13, and Figure 6.14 confirms this observation. All these figures show the three input frames, the computed flow field and an error visualization for one sequence of the KITTI 2015 benchmark and two sequences of the MPI Sintel benchmark, respectively. While Figure 6.12 makes the quantitative gains for the KITTI benchmark explicit, Figure 6.13 and Figure 6.14 show that our approach also allows obtaining improvements in case of non-rigid motion (fingers) and illumination changes (head of the dragon).

Only Prediction To further investigate the quality of the predicted flow fields, we performed a second experiment, where we skipped the combination step and only used the predicted flow to compute the final flow estimate. Thereby we computed the final estimate in two ways: once by solely inpainting the predicted flow field, i.e., without refinement, and once with the entire pipeline, i.e., with inpainting and refinement. We listed the outcome in Table 6.7 (only prediction). As one can see, in case of the KITTI 2012 and the KITTI 2015 benchmark, the pure prediction variant even outperforms our baseline. This observation not only confirms the high quality and reliability of our learned motion models but also reveals that due to the dominating forward motion in the benchmark many occlusions appear at the image boundaries and hence can be resolved by

Table 6.7: Results for the training data sets of the KITTI 2012 benchmark [65], the KITTI 2015 benchmark [119] and the MPI Sintel benchmark [44] (clean render path) in terms of the average endpoint error (AEE) and the percentage of bad pixels (BP) with a 3px threshold.

method		KITTI 2012		KITTI 2015		Sintel
	model	AEE	BP	AEE	BP	AEE
baseline	–	2.61 px	8.98 %	6.82 px	18.70 %	1.95 px
only prediction						
without refinement	constant	7.99 px	57.13 %	12.81 px	52.19 %	5.32 px
with refinement	constant	7.07 px	45.07 %	12.23 px	46.15 %	4.97 px
without refinement	learned	2.27 px	7.79 %	5.87 px	17.42 %	2.93 px
with refinement	learned	**1.83** px	**7.44** %	**5.37** px	**16.98** %	**2.29** px
full model						
our approach	constant	4.07 px	16.33 %	8.53 px	23.23 %	2.82 px
our approach	learned	**1.89** px	**7.26** %	**5.22** px	**16.25** %	**1.78** px
recent literature						
PWC-Net [163]	CVPR '18	4.14 px	–	10.35 px	33.67 %	2.55 px
FlowNet2 [89]	CVPR '17	4.09 px	–	10.06 px	30.37 %	2.02 px
UnFlow [118]	AAAI '18	**3.29** px	–	**8.10** px	23.27 %	–
DCFlow [198]	CVPR '17	–	–	–	15.09 %	–
MR-Flow [196]	CVPR '17	–	–	–	14.09 %	**1.83** px
Mirror Flow [87]	ICCV '17	–	–	–	**9.98** %	–
learning approaches (fine tuned)						
PWC-Net-ft[163]	CVPR '18	(1.45 px)	–	(2.16 px)	(9.80 %)	(1.70 px)
FlowNet2-ft [89]	CVPR '17	(1.28 px)	–	(2.30 px)	(8.61 %)	(1.45 px)
UnFlow-ft [118]	AAAI '18	(**1.14** px)	–	(**1.86** px)	(**7.40** px)	–

considering information from the preceding frame. The more challenging MPI Sintel benchmark, in contrast, does not contain such a high regularity of the present motion. Nevertheless, also, in this case, the learned prediction is still able to achieve reasonable results. For the sake of completeness, we also computed predictions based on the constant motion model. However, as one can see, the constant model does not allow to achieve nearly as good results as the learned approach.

Comparison to the Literature In our second to last experiment, we compare the performance of our novel optical flow approach to other methods from the literature. To this end, we consider both the training and the test data sets. In case of the training data we added results of recent methods to Table 6.7 (recent literature). While, our method generally yields better results than recent learning approaches without fine-tuning (PWC-Net [163], FlowNet2 [89], UnFlow [118]) and even outperforms all other approaches on the MPI Sintel benchmark, it scores slightly worse compared to the non-learning based methods on the KITTI 2015 benchmark. Regarding the test data sets, we submitted results to all three benchmarks. The results are shown in Table 6.8 and Table 6.9, where we have listed the ten best performing non-anonymous optical flow methods

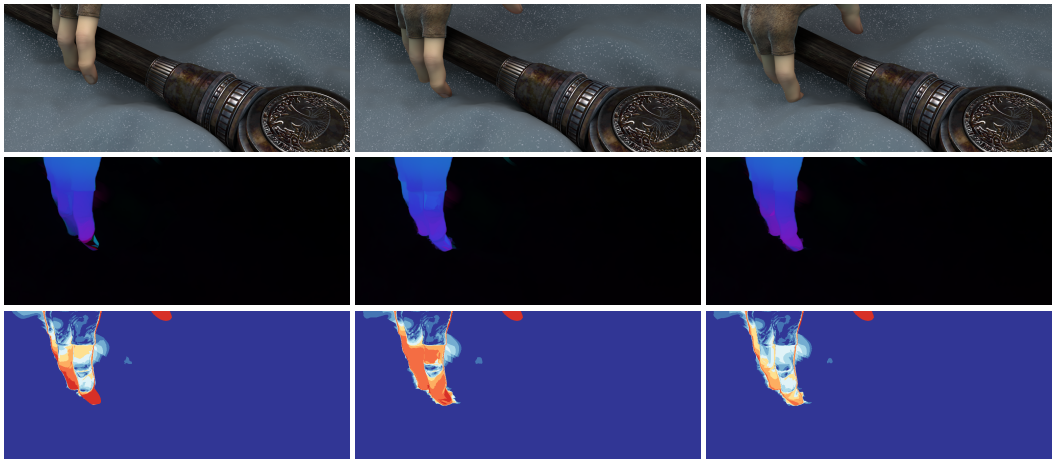


Figure 6.13: Improvements for non-rigid motion (MPI Sintel benchmark [44], ambush7 #9). *First row:* Previous, reference and subsequent frame. *Second and third row:* Estimated flow field, bad pixel visualization. *From left to right:* Baseline, constant motion model and our approach.

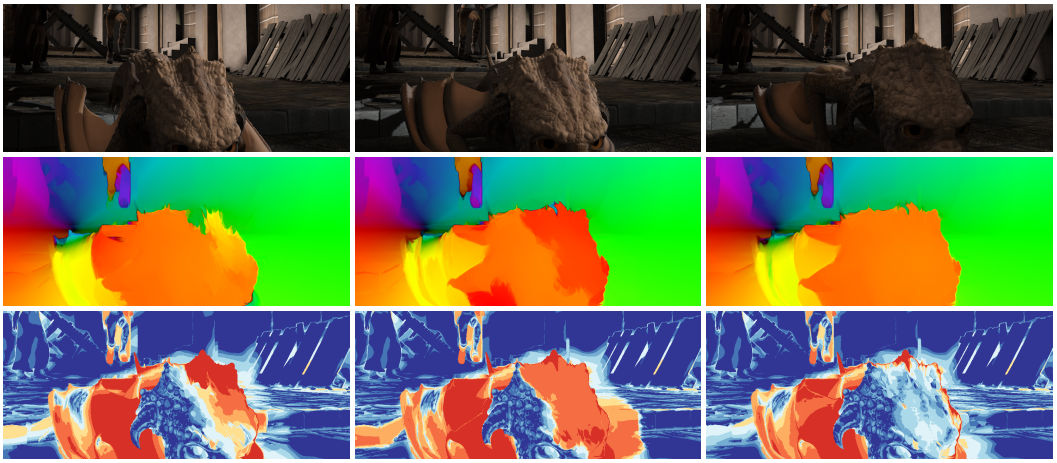


Figure 6.14: Improvements in case of illumination changes (MPI Sintel benchmark [44] market5 #8). *First row:* Previous, reference and subsequent frame. *Second and third row:* Estimated flow field, bad pixel visualization. *From left to right:* Baseline, constant motion model and our approach.

at the time of submission (Apr. 2018) for each benchmark. In case of KITTI 2012 our approach ranks the eighth w.r.t. the bad pixel error accounting only for pixels in non-occluded areas (Out-Noc). Since our method aims at improving the estimation in occluded areas, however, the bad pixel measure considering all pixels (Out-All) is more informative. Here, our approach ranks second. In case of the more challenging KITTI 2015 benchmark, we also rank eighth. However, on the most challenging and diverse benchmark, the MPI Sintel benchmark, we rank first in the final and second in the clean render path. In particular, in the significantly more challenging final render path, we

Table 6.8: Top 10 non-anonymous optical flow methods and methods presented in this thesis on the test data of the KITTI 2012 benchmark [65] and the KITTI 2015 benchmark [119] at time of submission (Apr. 2018), excluding scene flow methods. The presented method is highlighted in red. The methods presented in Section 6.1 (rigid motion model), Chapter 5 (DF+OIR), Section 4.1 (SODA-Flow) and Section 4.2 (OAR-Flow) are highlighted in blue.

KITTI 2012	Out-Noc	Out-All	Avg-Noc	Avg-All	KITTI 2015	Fl-bg	Fl-fg	Fl-all
SPS-Fl[1] [200]	3.38 %	10.06 %	0.9 px	2.9 px	PWC-Net [163]	9.66 %	9.31 %	9.60 %
PCBP-Flow[1] [201]	3.64 %	8.28 %	0.9 px	2.2 px	MirrorFlow [87]	8.93 %	17.07 %	10.29 %
SDF[2] [19]	3.80 %	7.69 %	1.0 px	2.3 px	SDF[2] [19]	8.61 %	23.01 %	11.01 %
MotionSLIC[1] [201]	3.91 %	10.56 %	0.9 px	2.7 px	UnFlow [118]	10.15 %	15.93 %	11.11 %
PWC-Net [163]	4.22 %	8.10 %	0.9 px	1.7 px	CNNF+PMBP [212]	10.08 %	18.56 %	11.49 %
UnFlow [118]	4.28 %	8.42 %	0.9 px	1.7 px	MR-Flow[2] [196]	10.13 %	22.51 %	12.19 %
MirrorFlow [87]	4.38 %	8.20 %	1.2 px	2.6 px	DCFlow [198]	13.10 %	23.70 %	14.86 %
our approach	**4.49** %	**7.88** %	**1.1** px	**2.1** px	our approach	**13.86** %	**20.91** %	**15.04** %
ImpPB+SPCI [154]	4.65 %	13.47 %	1.1 px	2.9 px	SOF[2] [156]	14.63 %	22.83 %	15.99 %
CNNF+PMBP [212]	4.70 %	14.87 %	1.1 px	3.3 px	JFS[2] [86]	15.90 %	19.31 %	16.47 %
rigid motion model	4.02 %	6.15 %	1.0 px	1.5 px	rigid motion model	9.66 %	22.73 %	11.83 %
DF+OIR [8]	5.17 %	10.43 %	1.1 px	2.9 px	DF+OIR [8]	15.11 %	23.45 %	16.50 %
SODA-Flow [10]	5.57 %	10.71 %	1.3 px	2.8 px	SODA-Flow [10]	20.01 %	29.14 %	21.53 %
OAR-Flow [9]	5.69 %	10.72 %	1.4 px	2.8 px	OAR-Flow [9]	20.62 %	27.67 %	21.79 %

[1] uses epipolar geometry as a hard constraint, only applicable to pure ego-motion
[2] exploits semantic information

Table 6.9: Top 10 non-anonymous optical flow methods and methods presented in this thesis on the test data of the MPI Sintel benchmark [44] at time of submission (Apr. 2018). The presented method is highlighted in red. The methods presented in Section 6.1 (rigid motion model), Chapter 5 (DF+OIR), and Section 4.2 (OAR-Flow) are highlighted in blue.

MPI Sintel final	all	matched	unmatched	MPI Sintel clean	all	matched	unmatched
our approach	**5.017** px	**2.596** px	**24.736** px	MR-Flow[2] [196]	2.527 px	0.954 px	15.365 px
PWC-Net [163]	5.042 px	2.445 px	26.221 px	our approach	**2.818** px	**1.027** px	**17.428** px
DCFlow [198]	5.119 px	2.283 px	28.228 px	FlowFields+ [21]	3.102 px	0.820 px	21.718 px
FlowFieldsCNN [22]	5.363 px	2.303 px	30.313 px	CPM2 [111]	3.253 px	0.980 px	21.812 px
MR-Flow[2] [196]	5.376 px	2.818 px	26.235 px	MirrorFlow [87]	3.316 px	1.338 px	19.470 px
S2F-IF [203]	5.417 px	2.549 px	28.795 px	DF+OIR [8]	3.331 px	0.942 px	22.817 px
InterpoNet_ff [219]	5.535 px	2.372 px	31.296 px	S2F-IF [203]	3.500 px	0.988 px	23.986 px
RicFlow [84]	5.620 px	2.765 px	28.907 px	SPM-BPv2 [110]	3.515 px	1.020 px	23.865 px
InterpoNet_cpm [219]	5.627 px	2.594 px	30.344 px	DCFlow [84]	3.537 px	1.103 px	23.394 px
ProbFlowFields [181]	5.696 px	2.545 px	31.371 px	RicFlow [84]	3.550 px	1.264 px	22.220 px
rigid motion model	5.466 px	2.683 px	28.147 px	rigid motion model	2.910 px	1.016 px	18.357 px
DF+OIR [8]	5.862 px	2.864 px	30.303 px	–	–	–	–
OAR-Flow [9]	8.179 px	4.578 px	37.525 px	OAR-Flow [9]	6.227 px	2.760 px	34.455 px

[1] uses epipolar geometry as a hard constraint, only applicable to pure ego-motion
[2] exploits semantic information

not only obtain the best result but also obtain the lowest error in occluded areas (unmatched) – even outperforming modern multi-frame methods such as MR-Flow [196] that combine geometric constraints with a semantic rigidity segmentation. These results show that in particular in difficult scenes with partially non-rigid motion, learned temporal models might be a worthwhile strategy.

Table 6.10: Optical flow leaderboard of the Robust Vision Challenge (Jun. 2018). Numbers in parentheses denote the rank on the respective benchmark w.r.t. all published methods based on the default error measures.

method	overall rank	Middlebury [23]	KITTI [119]	Sintel [44]	HD1K [103]
PWC-Net_ROB [163, 165]	1	2 (35)	2 (11)	2 (2/ 32)	1
our approach	**2**	**1** (14)	**5** (18)	**1** (3/ 2)	**3**
LFNet_ROB	3	6 (106)	1 (8)	5 (34/ 54)	4
AugFNG_ROB	4	8 (109)	3 (27)	3 (17/ 21)	2
FF++_ROB [153]	4	3 (64)	4 (19)	4 (49/ 35)	5
DMF_ROB [191]	6	4 (79)	7 (66)	6 (68/ 63)	7
ResPWCR_ROB	6	5 (88)	6 (24)	7 (50/ 69)	6
WOLF_ROB	8	7 (105)	8 (76)	8 (97/102)	8
TVL1_ROB	9	9 (116)	9 (80)	9 (125/121)	9
H+S_ROB	10	10 (134)	10 (83)	10 (137/135)	10
# methods	–	10 (151)	10 (87)	10 (144/144)	10

Robust Vision Challenge The increasing availability of benchmarks has not only lead to tremendous progress in computer vision but has also enabled us to compare the results of dozens of methods easily. However, often this steady progress is made on each individual benchmark, i.e., it is limited to a specific domain/benchmark, and the state-of-the-art methods often do not perform well on different datasets without a substantial adaption of the model parameters. To tackle this issue and foster the development of algorithms that are robust and perform well on a variety of diverse datasets the Robust Vision Challenge[1] was held in June 2018. The task of the optical flow category was to apply a method using the same parameter setting/model to four different benchmarks: the MPI Sintel benchmark [44], the KITTI 2015 benchmark [119], the HD1K benchmark [103], and the Middlebury benchmark [23].

To participate in the challenge, we only had to choose a fixed set of parameters for the refinement, since the other parameters already had been fixed across all benchmarks. Hence, we computed a parameter set by minimizing the BP error on a subset of the provided training data. Table 6.10 shows the final leaderboard of the Robust Vision Challenge. As one can see, we ranked second place with the best performance on the MPI Sintel and Middlebury benchmark. Solely in the two automotive benchmarks, i.e., KITTI 2015 and HD1K benchmark, which exhibit a high regularity, we trail the end-to-end learning approaches. Furthermore, we also listed the rank of the methods on the respective benchmarks w.r.t. all published methods (based on the default error measures) in parentheses. These results demonstrate nicely that our presented approach yields an excellent performance across different domains without the need for a specific parameter adaptation.

Runtime Running our approach on a desktop PC equipped with an Intel Core i7-7820X CPU @ 3.60GHz and an Nvidia GeForce GTX 1070 the runtime is approximately 112s for a flow field of size 1226×370. The overall runtime splits up into: 36s for the initial flow field estimation, 50s for the motion model learning (CNN training) and prediction, and another 26s for the final inpainting and refinement.

[1] http://www.robustvision.net

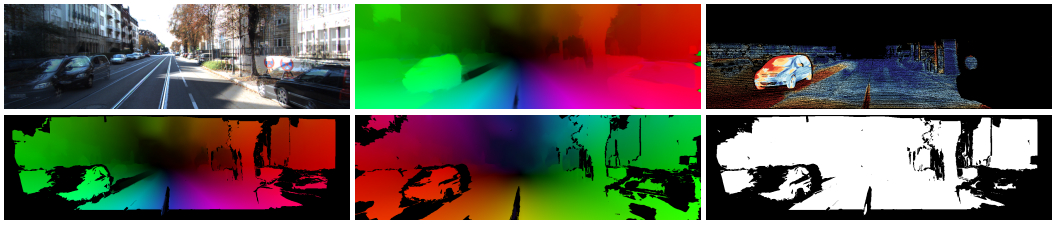


Figure 6.15: Limitations example (KITTI 2015 benchmark [119], seq. #0). *First row:* Overlayed reference and subsequent input frame, final flow estimate, bad pixel visualization. *Second row:* Filtered forward flow, filtered backward flow, possible training candidates (white).

6.2.5 Limitations

Finally, we also want to comment on the limitations of our approach. In the case of large image regions that only contain poor or possibly no training samples, the validity of the learned motion model may not be able to generalize to the entire image domain. In Figure 6.15 such a scenario is depicted. Due to the missing training samples at the bottom corners of the image, the prediction cannot achieve a noticeable improvement in these areas. This problem, however, could be resolved by additionally using geometric constraints in terms of a rigid motion model. Hence, we believe that combining our learning based approach with such a model could even allow for further improvements – at least in case of rigid scenes with a vast amount of ego-motion, such as the KITTI 2012 and the KITTI 2015 benchmark.

6.2.6 Conclusions

In this section, we presented a novel multi-frame optical flow approach that integrates flow predictions based on a CNN. To this end, we made use of an unsupervised/self-supervised learning approach that learns a motion model by estimating a spatially variant mapping from the backward to the forward flow. In contrast to existing approaches from the literature that train their network only once before the estimation based on a vast data set, our method exploits flow estimates from the current image sequence to learn the model online, i.e., during the estimation. In this way, it becomes possible to learn motion models that are specifically tailored to the actual motion occurring in each frame. Experiments made a good performance of our method explicit. They not only show significant improvements compared to a baseline without prediction, but they also show consistently good results in all major benchmarks – including top results on the Sintel benchmark.

6.3 Summary

In this chapter, we introduced two new approaches for motion estimation that go beyond the classical two-frame setting. To this end, we proposed the use of two different motion models: a rigid-motion model and a learned motion model. While both motion models allowed to significantly improve the estimation accuracy compared to the two-frame setting, they both have individual advantages and disadvantages. Hence, we briefly summarize our findings. Not surprisingly, the rigid-motion model proves to be the ideal model in case of ego-motion scenes, which contain a high share of pixels depicting the background (e.g., KITTI 2012). However, no benefits can be

achieved in case of non-rigid motion and in scenes with a large amount of independent moving objects. In contrast, the learned-motion model allows to obtain improvements for a great variety of different scenes, including scenes that contain non-rigid motion and independently moving objects (e.g., MPI Sintel). However, regarding ego-motion scenes the benefits are, in fact, slightly lower compared to the ego-motion model.

7 Conclusions

7.1 Conclusions

In this thesis, we have considered two fundamental problems of computer vision: 3D reconstruction and motion estimation. In particular we have investigated and contributed several novel ideas that advanced the field, which we will summarize in the following paragraphs.

In Chapter 3, we developed a variational method that simultaneously leverages shading and parallax information to reconstruct a static object or scene. In contrast to other methods, we demonstrated that it is possible to integrate both sources of information into a joint minimization framework that does not require any kind of pre-estimation. Furthermore, we designed the underlying model in such a way that it not only estimates the depth but also the surface albedo as well as the present illumination. This design choice enables our approach to deal with a broad variety of different scenarios involving Lambertian objects with non-uniform albedo and unknown illumination settings. To implement the proposed model we derived a coarse-to-fine minimization framework based on a linearization of all data terms. This linearization not only enabled the application of standard optimization techniques such as nested fixed point iterations, but it also allowed the joint estimation of all unknowns. Finally, our experiments considering synthetic as well as real-world images demonstrate that our new combined approach allows for accurate and detailed reconstructions. Moreover, they show that shading cues are indeed useful to improve upon pure parallax based methods, in particular when it comes to the reconstruction of small-scale details.

In Chapter 4 we turned to the topic of motion estimation. In the first part of this chapter, we focused on *second-order regularization techniques* for variational motion estimation. Besides exploring several different modeling strategies to realize second-order regularization, we also demonstrated how to include a directional-depended (anisotropic) smoothing behavior within the regularization process. In this context, we showed that modeling a higher degree of anisotropy in terms of a double anisotropic model can further improve the result in terms of quality. To this end, we ran several experiments on the KITTI 2012 and KITTI 2015 benchmark to quantitatively compare all the different regularization techniques with each other. Further comparisons with variational approaches from the literature revealed that our new double anisotropic second-order coupling model achieves state-of-the-art results in the context of variational motion estimation.

In the second part of this chapter, we addressed a common drawback of second-order regularizers, i.e., the fact that they are less suited to estimate fronto-parallel motion compared to first-order regularizer, since they are likely to misinterpret local fluctuations as affine motion. In this context, we proposed an *order-adaptive regularization* strategy that automatically adapts the utilized regularization order. To steer the underlying adaption process we designed four different adaptation schemes: a global (per-frame considering all locations), a local (per-location considering a single location), a non-local (per-location considering a small-neighborhood), and a region-based scheme

(per-location considering a single location and smoothness). While the global adaptation strategy turned out to be highly robust at the expense of being less adaptive, the local approach allowed a flexible point-wise selection at the cost of producing noisy decisions. By imposing some form of spatial regularity, i.e., neighborhood information or a spatial smoothness term, we succeeded to combine the advantages of both strategies. Finally, we confirmed these considerations in our experiments. They showed that adaptively combining different regularization orders not only allows outperforming the non-adaptive strategy but also that a location-wise adaptivity may turn out useful if we regularize the decision process.

In Chapter 5, we moved from entirely variational methods to pipeline-based approaches. Typically, such pipeline-based methods employ a fairly simplistic variational refinement that is only capable to achieve minor improvements. To tackle this shortcoming we proposed a novel variational refinement scheme that combines an illumination-aware data term with our new order-adaptive regularization scheme. While the choice of data term allows the new model to keep up with many feature descriptors, our order-adaptive regularization term allows the new model to deal with more complex motion patterns. Besides the novel refinement model, we also proposed a hierarchical refinement scheme that starts the computation at an intermediate resolution level. This choice allows the variational refinement to benefit from a good initialization while still being able to correct errors. Finally, consistently good results on popular optical flow benchmarks showed that our novel variational refinement strategy not only allows to improve outcomes compared to traditional refinement schemes but also that it allows outperforming pure variational methods.

In the last chapter (Chapter 6) we developed strategies to exploit information on temporal coherence by utilizing information from additional preceding input frames. These strategies not only allowed us to increase the robustness, but also remarkably improve the estimation within occluded areas. To realize them we employed motion models that allow to relate the sought displacement vector field to motion estimates from the past. In particular, we made use of two different models: a rigid-motion model and a learned motion model.

In the first part of this chapter, we proposed a multi-frame approach that builds upon the rigid-motion model. The method incorporates information from additional frames by integrating structure information. To this end, we developed a hierarchical depth-parametrized three-frame SfM/stereo PatchMatch approach with temporal selection and preceding pose estimation. Further, we introduced a consistency based combination scheme that allows to combine the resulting structure matches with those of a recent PatchMatch approach for general motion estimation without the need of any semantic information. Experiments not only showed excellent results on all major benchmarks, i.e., the KITTI 2012, KITTI 2012 and MPI Sintel benchmark, they also demonstrated consistent improvements over a baseline without structure information.

In the second part of this chapter, we presented a novel multi-frame method that builds upon a learned motion-model. To this end, we proposed a self-supervised convolutional neural network that learns a motion model in terms of a spatially variant mapping from the backward to the forward flow. Furthermore, we pursued an online training approach, i.e., the training process is performed during the estimation, and solely relies on flow estimates from the current image sequence for training. This choice enabled us to learn motion models that are specifically tailored to the actual motion occurring in each frame. Experiments on all major benchmarks made the great performance of our method explicit, most notably achieving the top ranking on the MPI Sintel

benchmark at time of submission. Further we were able to score the runner-up award within the CVPR 2018 Robust Vision Challenge with the best performance on the MPI Sintel and Middlebury benchmark, thereby demonstrating the generalization capability of our new approach.

In summary, we can say that we advanced the field of computer vision by providing valuable contributions to the topics of 3D reconstruction and motion estimation. In particular, regarding the adaptivity of models to the underlying data and thereby improving in terms of general applicability to a broad variety of possible input data.

7.2 Future Work

Although we were able to provide useful insights and develop highly accurate techniques for 3D reconstruction and motion estimation, plenty of challenges remain unsolved. Hence, in the following, we discuss some promising possibilities concerning future work.

3D Reconstruction Our new approach presented in Chapter 3 already copes with complex lighting scenarios and recovers fine surface details. However, it reaches its limits in case of shiny objects. Consequently, it would be desirable to extend the model assumptions to non-Lambertian surfaces to deal with more complex object materials [95, 117, 178]. This extension would allow the method to be applicable to a greater variety of objects. Another aspect that could be generalized is the need of a pre-calibrated camera setup. While our approach does not require any lighting calibration, it assumes the camera poses to be known, which are typically estimated in a pre-processing step. Hence, it would be worthwhile to investigate the possibility to further jointly estimate the extrinsic camera calibration [169, 214]. This extension would render our method to be even easier applicable, i.e., no re-calibration is would be needed if the capturing viewpoints are altered.

Motion Estimation In Chapter 6 we proposed two different approaches to realize multi-frame motion estimation. While the first strategy based on the rigid-motion model excelled in case of ego-motion dominant sequences with a rigid background, the second strategy based on the learned motion model achieved great improvements even in case of non-rigid motion sequences. Therefore, it could be rewarding to combine both strategies within a joint approach and maximize the improvements for a variety of different sequences. Furthermore, the employed components do not use the available information to full capacity, i.e., areas that have been identified as possibly occluded are not treated as such in the subsequent refinement steps. Hence, by using all information gathered in the pipeline in the final steps, in particular in the refinement step, could lead to even further benefits.

Unsupervised Learning A recent trend in the context end-to-end learning methods for motion estimation is to resort to an unsupervised training procedure [92, 118], i.e., training without the need of labeled training data. This not only enables the approaches to eliminate the costly part of acquiring labels for real-world training data but typically also enables the use of a virtually indefinite amount of training data. To realize this unsupervised training, such approaches typically employ an unsupervised loss-function during training. These loss functions can basically be equated with the energy functionals employed in this thesis. However, in contrast to minimizing the loss during estimation, end-to-end learning methods minimize the loss during training to fix the large amount of parameters of the underlying CNN. Consequently, our new models in

Chapter 4 and Chapter 5 can be directly applied as unsupervised loss-functions. Therefore, it could be worthwhile analyzing the benefits of using our advanced models compared to the so far employed loss functions. In this context one could also incorporate our ideas for multi-frame motion estimation and replace simple constant motion models [92] with more advanced motions models, e.g., the rigid-motion model or a learned motion model as proposed in Chapter 6.

3D Reconstruction and 3D Motion Estimation Another interesting path of future research would be to combine our ideas from both topics, i.e., 3D reconstruction and motion estimation, in order to realize an approach that not only estimates the 3D geometry of the scene but also computes the 3D motion of dynamic objects within this scene – the so-called *scene flow* problem [26, 172, 173]. To this end, one could leverage multi-frame information in terms of varying viewpoints and varying points in time. Furthermore, reasoning about the 3D geometry, the 3D motion, the reflectance properties, and the present illumination could allow us to employ more sophisticated priors and tackle various challenges, e.g., illumination changes. While many research in this area addresses an automotive context [119, 183], this type of additional information, i.e., reflectance properties and illumination information, might be of particular interest in non-automotive contexts such as augmented and virtual reality.

Runtime Performance Finally, another important aspect is the runtime performance. While we have not focused our attention on this topic, it becomes a crucial point when dealing with time-critical or even real-time applications. In order to tackle this challenge, one can consider using more sophisticated numerical schemes for variational methods, e.g., multigrid techniques [39, 41, 42], or employing other numerical solvers that allow to use the highly parallel structure of nowadays hardware such as GPUs, e.g. advanced explicit schemes [70, 74, 76, 190], primal-dual methods [46, 104, 210] or domain decomposition methods [101, 102]. In this context, one can also highlight that such highly parallel hardware enable other recent approaches to achieve good runtime performances, i.e., randomized approaches [62] and CNN based learning approaches [88, 139, 163]. This in turn opens up the possibility of replacing individual components from the presented pipeline approaches by CNNs to achieve an improvement in runtime performance.

A Details and Derivations

A.1 Linearization Stereo Data Term

In this section, we provide additional information regarding the implementation of the differential stereo data term. For the sake of clarity we focus on the brightness constancy assumption of a single color channel and drop the iteration index k, this gives us

$$\varphi_i(\mathbf{x}) := I_0(\mathbf{x}) - I_i(\mathbf{x_i}) \,. \tag{A.1}$$

For which the corresponding linearized expression reads

$$\bar{\varphi}_i(\mathbf{x}) := \varphi_i(\mathbf{x}) + \partial_z \varphi_i(\mathbf{x}) \cdot dz(\mathbf{x}) \,. \tag{A.2}$$

On the one hand, this linearized expression includes $I_i(\mathbf{x_i})$, which one can realize with the principle of warping, similar to the optical flow example model in Section 2.4.2. On the other hand, it requires to compute the derivative of $\varphi_i(\mathbf{x})$ w.r.t. the depth ∂_z, which we have not addressed so far. Below, we detail on the implementation of both expressions.

A.1.1 Stereo Warping

In order to compute expressions like $I_i(\mathbf{x_i})$, we warp the i-th image towards the reference view captured by $\mathbf{C}_0$. We can interpret this step as a depth compensation by the intermediate depth estimate z. In contrast to our optical flow example, we cannot directly obtain the corresponding location by adding the estimated displacement. We first have to compute the corresponding 3D surface point $\mathbf{s}(\mathbf{x}, z)$ and project it via $\pi_i(\mathbf{s}(\mathbf{x}, z))$ to obtain the desired location $\mathbf{x}_i$. The warped image $I_i^{\mathbf{w}}$ writes the brightness values of the corresponding locations to the current locations $I_i^{\mathbf{w}}(\mathbf{x}) = I_i(\mathbf{x_i})$. In case of a discrete implementation, $\mathbf{x_i}$ will lie, in most cases, between the sampled locations and the actual value must be approximated, e.g., using bilinear interpolation.

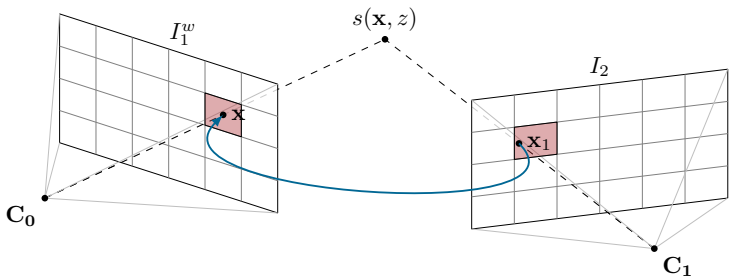


Figure A.1: Sketch showing the basic implementation of warping (stereo case).

A.1.2 Depth Derivatives

Next, we take a look at how to compute the derivatives w.r.t. the depth, i.e., the derivative of $\varphi_i(\mathbf{x})$ w.r.t. ∂_z. Applying the chain rules results in

$$\begin{aligned} \partial_z \varphi_i(\mathbf{x}) &= \partial_z \varphi_i(\mathbf{x}) && \text{(A.3)} \\ &= \underbrace{\partial_z I_0(\mathbf{x})}_{=0} - \partial_z I_i(\mathbf{x_i}) && \text{(A.4)} \\ &= -\nabla_i I_i(\mathbf{x_i})^\top \partial_z \mathbf{x_i} && \text{(A.5)} \end{aligned}$$

where the ∇_i operator is defined as $\nabla_i = (\partial_{x_i}, \partial_{y_i})^\top$. Furthermore, we recall that $\mathbf{x_i}$ is given by

$$\mathbf{x_i} = \pi(K_i \left(R_i \left(z \cdot K_0^{-1} \tilde{\mathbf{x}}\right) + \mathbf{t}_i\right), \tag{A.6}$$

hence applying the chain rule to $\partial_z \mathbf{x_i}$ results in

$$\partial_z \mathbf{x_i} = \underbrace{\mathcal{J}(\pi)}_{\text{Jacobian of } \pi} \quad \underbrace{K_i \, R_i \, K_0^{-1} \tilde{\mathbf{x}}}_{\text{inner derivative}}. \tag{A.7}$$

By introducing the following abbreviations for $\mathbf{a}$ and $\mathbf{b}$

$$K_i \left(R_i \left(z \cdot K_0^{-1} \tilde{\mathbf{x}}\right) + \mathbf{t}_i\right) = \underbrace{K_i \, R_i \, K_0^{-1} \tilde{\mathbf{x}}}_{\mathbf{a}} \cdot z + \underbrace{K_i \, \mathbf{t}_i}_{\mathbf{b}} = \mathbf{x} \cdot z + \mathbf{b} = \begin{pmatrix} a_0 \cdot z + b_0 \\ a_1 \cdot z + b_1 \\ a_2 \cdot z + b_2 \end{pmatrix}, \tag{A.8}$$

the derivative can be written more explicit as

$$\partial_z \mathbf{x_i} = \begin{pmatrix} \frac{1}{a_2 \cdot z + b_2} & 0 & -\frac{a_0 \cdot z + b_0}{(a_2 \cdot z + b_2)^2} \\ 0 & \frac{1}{a_2 \cdot z + b_2} & -\frac{a_1 \cdot z + b_1}{(a_2 \cdot z + b_2)^2} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} \frac{a_0 b_2 - b_0 a_2}{(a_2 \cdot z + b_2)^2} \\ \frac{a_1 b_2 - b_1 a_2}{(a_2 \cdot z + b_2)^2} \end{pmatrix}. \tag{A.9}$$

Finally, the last missing part of Equation A.3 is the computation of $\nabla_i I_i(\mathbf{x_i})$. We can either follow [116] and compute the spatial gradients on the second image

$$\nabla_i I_i^{\mathbf{w}}(\mathbf{x}) = \nabla_i I_i(\mathbf{x_i}) \tag{A.10}$$

or follow [26] and compute the gradients of the warped image and relate them via the Jacobian

$$\left(\nabla I_i^{\mathbf{w}}(x, y)\right)^\top = \left(\nabla I_i(\mathbf{x_i})\right)^\top = \nabla_i I_i(\mathbf{x_i})^\top \mathcal{J}(\mathbf{x_i}) = \nabla_i I_i(\mathbf{x_i})^\top \begin{pmatrix} \frac{\partial x_i}{\partial x} & \frac{\partial x_i}{\partial y} \\ \frac{\partial y_i}{\partial x} & \frac{\partial y_i}{\partial y} \end{pmatrix}, \tag{A.11}$$

where the entries of the Jacobian $\mathcal{J}(\mathbf{x_i})$ are given by

$$\partial_x \mathbf{x_i} = \begin{pmatrix} \partial_x x_i \\ \partial_x y_i \end{pmatrix} = \mathcal{J}(\pi) \, K_i R_i K_0^{-1} (z_x \tilde{\mathbf{x}} + z \, \mathbf{e}_1), \tag{A.12}$$

$$\partial_y \mathbf{x_i} = \begin{pmatrix} \partial_y x_i \\ \partial_y y_i \end{pmatrix} = \mathcal{J}(\pi) \, K_i R_i K_0^{-1} (z_y \tilde{\mathbf{x}} + z \, \mathbf{e}_2). \tag{A.13}$$

A.2 Fourth Order Diffusion Tensor

The direct second-order regularization leads to a fourth order diffusion process. This diffusion process contains a 4×4 tensor that is considered the equivalent to the diffusion tensor from ordinary second order diffusion. Therefore, we refer to it as the fourth order diffusion tensor. Introducing the following abbreviation

$$\Psi'_{l,m} := \Psi'_{l,m}\left(\left(\mathbf{r}_m^\top \mathcal{H}u\,\mathbf{r}_l\right)^2 + \left(\mathbf{r}_m^\top \mathcal{H}v\,\mathbf{r}_l\right)^2\right), \tag{A.14}$$

where we dropped the argument of the penalizer functions, the tensor can be written as

$$T_{\text{2-aniso-d}} = \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot \left(\mathbf{r}_m \mathbf{r}_m^\top \otimes \mathbf{r}_l \mathbf{r}_l^\top\right) \tag{A.15}$$

Derivation To derive the fourth order diffusion tensor, we take a look at the Euler-Lagrange equations of the direct second-order regularizer. For u this is given by

$$0 = R_u - \frac{\partial}{\partial x}R_{u_x} - \frac{\partial}{\partial y}R_{u_y} + \frac{\partial^2}{\partial x \partial x}R_{u_{xx}} + \frac{\partial^2}{\partial x \partial y}R_{u_{xy}} + \frac{\partial^2}{\partial y \partial x}R_{u_{yx}} + \frac{\partial^2}{\partial y \partial y}R_{u_{yy}}\,, \tag{A.16}$$

where R denotes the double anisotropic second order regularizer as in Equation 4.13

$$R(\mathbf{w}) = \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi_{l,m}\left(\left(\mathbf{r}_m^\top \mathcal{H}u\,\mathbf{r}_l\right)^2 + \left(\mathbf{r}_m^\top \mathcal{H}v\,\mathbf{r}_l\right)^2\right). \tag{A.17}$$

While the contributions R_u, R_{u_x}, and R_{u_y} are zero, the remaining contributions are given by

$$R_{u_{xx}} = \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot \left(\mathbf{r}_m^\top \mathcal{H}_u\,\mathbf{r}_l\right) \cdot r_{m1} r_{l1}\,, \tag{A.18}$$

$$R_{u_{xy}} = \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot \left(\mathbf{r}_m^\top \mathcal{H}_u\,\mathbf{r}_l\right) \cdot r_{m1} r_{l2}\,, \tag{A.19}$$

$$R_{u_{yx}} = \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot \left(\mathbf{r}_m^\top \mathcal{H}_u\,\mathbf{r}_l\right) \cdot r_{m2} r_{l1}\,, \tag{A.20}$$

$$R_{u_{yy}} = \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot \left(\mathbf{r}_m^\top \mathcal{H}_u\,\mathbf{r}_l\right) \cdot r_{m2} r_{l2}\,, \tag{A.21}$$

where we dropped the arguments of the Ψ'_* functions to enhance the readability. By writing $\mathbf{x} = (x, y)$ as $\mathbf{x} = (x_1, x_2)$, we can parametrize the derivative directions by indexing and generalize the expressions above:

$$R_{u_{x_i x_j}} = \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot \left(\mathbf{r}_m^\top \mathcal{H}_u \mathbf{r}_l\right) \cdot r_{mi} r_{lj} \qquad \forall i, j \in \{1, 2\}\,. \tag{A.22}$$

The second factor $\mathbf{r}_m^\top \mathcal{H}_u \, \mathbf{r}_l$ evaluates as

$$\mathbf{r}_m^\top \mathcal{H}_u \, \mathbf{r}_l = \begin{pmatrix} r_{m1}, & r_{m2} \end{pmatrix} \begin{pmatrix} u_{x_1x_1} & u_{x_1x_2} \\ u_{x_2x_1} & u_{x_2x_2} \end{pmatrix} \begin{pmatrix} r_{l1} \\ r_{l2} \end{pmatrix} \tag{A.23}$$

$$= \begin{pmatrix} r_{m1}, & r_{m2} \end{pmatrix} \begin{pmatrix} u_{x_1x_1} r_{l1} + u_{x_1x_2} r_{l2} \\ u_{x_2x_1} r_{l1} + u_{x_2x_2} r_{l2} \end{pmatrix} \tag{A.24}$$

$$= r_{m1} u_{x_1x_1} r_{l1} + r_{m1} u_{x_1x_2} r_{l2} + r_{m2} u_{x_2x_1} r_{l1} + r_{m2} u_{x_2x_2} r_{l2} \tag{A.25}$$

$$= \begin{pmatrix} r_{m1} r_{l1}, & r_{m1} r_{l2}, & r_{m2} r_{l1}, & r_{m2} r_{l2} \end{pmatrix} \begin{pmatrix} u_{x_1x_1} \\ u_{x_1x_2} \\ u_{x_2x_1} \\ u_{x_2x_2} \end{pmatrix} \tag{A.26}$$

$$= (\mathbf{r}_m \otimes \mathbf{r}_l)^\top ((\nabla \otimes \nabla) u) \, , \tag{A.27}$$

which allows us to re-write Equation A.22 as

$$R_{u_{x_i x_j}} = \sum_{l=1}^{2} \sum_{m=1}^{2} \Psi'_{l,m} \cdot \Big((\mathbf{r}_m \otimes \mathbf{r}_l)^\top ((\nabla \otimes \nabla) u) \Big) \cdot r_{mi} r_{lj} \tag{A.28}$$

$$= \sum_{l=1}^{2} \sum_{m=1}^{2} \Psi'_{l,m} \cdot r_{mi} r_{lj} \cdot (\mathbf{r}_m \otimes \mathbf{r}_l)^\top ((\nabla \otimes \nabla) u) \, . \tag{A.29}$$

By defining $\nabla^2 = \nabla \otimes \nabla = (\partial_{x_1x_1}, \partial_{x_1x_2}, \partial_{x_2x_1}, \partial_{x_2x_2})^\top$ as a kind of second order nabla operator and using a corresponding second order divergence equivalent operator $\nabla^2 \cdot$ leads to

$$\frac{\partial^2}{\partial x_1 \partial x_1} R_{u_{x_1x_1}} + \frac{\partial^2}{\partial x_1 \partial x_2} R_{u_{x_1x_2}} + \frac{\partial^2}{\partial x_2 \partial x_1} R_{u_{x_2x_1}} + \frac{\partial^2}{\partial x_2 \partial x_2} R_{u_{x_2x_2}}$$
$$= \nabla^2 \cdot \begin{pmatrix} R_{u_{x_1x_1}}, & R_{u_{x_1x_2}}, & R_{u_{x_2x_1}}, & R_{u_{x_2x_2}} \end{pmatrix}^\top , \tag{A.30}$$

we can re-write the right side of the Euler-Lagrange equation as

$$\nabla^2 \cdot \left(T_{\text{2-aniso-d}} \nabla^2 u \right) . \tag{A.31}$$

Now, we want to derive the fourth order diffusion tensor $T_{\text{2-aniso-d}}$.

$$T_{\text{2-aniso-d}} \nabla^2 u = \begin{pmatrix} R_{u_{x_1x_1}}, & R_{u_{x_1x_2}}, & R_{u_{x_2x_1}}, & R_{u_{x_2x_2}} \end{pmatrix}^\top \tag{A.32}$$

$$= \begin{pmatrix} \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot r_{m1} r_{l1} (\mathbf{r}_m \otimes \mathbf{r}_l)^\top (\nabla^2 u) \\ \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot r_{m1} r_{l2} (\mathbf{r}_m \otimes \mathbf{r}_l)^\top (\nabla^2 u) \\ \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot r_{m2} r_{l1} (\mathbf{r}_m \otimes \mathbf{r}_l)^\top (\nabla^2 u) \\ \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot r_{m2} r_{l2} (\mathbf{r}_m \otimes \mathbf{r}_l)^\top (\nabla^2 u) \end{pmatrix} \tag{A.33}$$

$$= \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot \begin{pmatrix} r_{m1} r_{l1} \\ r_{m1} r_{l2} \\ r_{m2} r_{l1} \\ r_{m2} r_{l2} \end{pmatrix} (\mathbf{r}_m \otimes \mathbf{r}_l)^\top (\nabla^2 u) \tag{A.34}$$

$$= \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot (\mathbf{r}_m \otimes \mathbf{r}_l)(\mathbf{r}_m \otimes \mathbf{r}_l)^\top (\nabla^2 u)\,. \tag{A.35}$$

Hence, the fourth order diffusion tensor is given as

$$T_{\text{2-aniso-d}} = \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot (\mathbf{r}_m \otimes \mathbf{r}_l)(\mathbf{r}_m \otimes \mathbf{r}_l)^\top \tag{A.36}$$

$$= \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot (\mathbf{r}_m \otimes \mathbf{r}_l)(\mathbf{r}_m^\top \otimes \mathbf{r}_l^\top) \tag{A.37}$$

$$= \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot (\mathbf{r}_m \mathbf{r}_m^\top \otimes \mathbf{r}_l \mathbf{r}_l^\top)\,. \tag{A.38}$$

Block Structure Further, the tensor can be partitioned into four 2×2 blocks and written as the following block matrix

$$T_{\text{2-aniso-d}} = \begin{pmatrix} A & B \\ B & C \end{pmatrix}. \tag{A.39}$$

First of all, let us write the tensor using four different blocks

$$T_{\text{2-aniso-d}} = \begin{pmatrix} A & B \\ D & C \end{pmatrix}. \tag{A.40}$$

In order to clarify why we can partition $T_{\text{2-aniso-d}}$ this way, we rewrite Equation A.38 as

$$T_{\text{2-aniso-d}} = \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot \left(\mathbf{r}_m \mathbf{r}_m^\top \otimes \mathbf{r}_l \mathbf{r}_l^\top\right) \tag{A.41}$$

$$= \sum_{l=1}^{2}\sum_{m=1}^{2} \Psi'_{l,m} \cdot \left(\begin{pmatrix} r_{m1} r_{m1} & r_{m1} r_{m2} \\ r_{m2} r_{m1} & r_{m2} r_{m2} \end{pmatrix} \otimes \mathbf{r}_l \mathbf{r}_l^\top\right). \tag{A.42}$$

By considering the construction properties of the Kronecker product, each of the four blocks is given as the sum of the 2×2 matrices $\mathbf{r}_l \mathbf{r}_l^\top$ weighted by $\Psi'_{l,m}$ and one of the entries of $\mathbf{r}_m \mathbf{r}_m^\top$:

$$A = \sum_{l=1}^{2} \sum_{m=1}^{2} \Psi'_{l,m} \cdot r_{m1} r_{m1} \cdot \mathbf{r}_l \mathbf{r}_l^\top , \tag{A.43}$$

$$B = \sum_{l=1}^{2} \sum_{m=1}^{2} \Psi'_{l,m} \cdot r_{m1} r_{m2} \cdot \mathbf{r}_l \mathbf{r}_l^\top , \tag{A.44}$$

$$D = \sum_{l=1}^{2} \sum_{m=1}^{2} \Psi'_{l,m} \cdot r_{m2} r_{m1} \cdot \mathbf{r}_l \mathbf{r}_l^\top , \tag{A.45}$$

$$C = \sum_{l=1}^{2} \sum_{m=1}^{2} \Psi'_{l,m} \cdot r_{m2} r_{m2} \cdot \mathbf{r}_l \mathbf{r}_l^\top . \tag{A.46}$$

Considering the commutativity of the scalar multiplication, it is now easy to see that $B = D$ holds. In the single anisotropic case, the diffusion tensor exhibits a block diagonal structure

$$T_{\text{2-aniso-s}} = \sum_{l=1}^{2} \sum_{m=1}^{2} \Psi'_l \cdot \left(\mathbf{e}_m \mathbf{e}_m^\top \otimes \mathbf{r}_l \mathbf{r}_l^\top \right) \tag{A.47}$$

$$= \sum_{l=1}^{2} \left(\Psi'_l \cdot \sum_{m=1}^{2} \left(\mathbf{e}_m \mathbf{e}_m^\top \otimes \mathbf{r}_l \mathbf{r}_l^\top \right) \right) \tag{A.48}$$

$$= \sum_{l=1}^{2} \left(\Psi'_l \cdot \left(\left(\sum_{m=1}^{2} \mathbf{e}_m \mathbf{e}_m^\top \right) \otimes \mathbf{r}_l \mathbf{r}_l^\top \right) \right) \tag{A.49}$$

$$= \sum_{l=1}^{2} \left(\Psi'_l \cdot \left(I_{2\times 2} \otimes \mathbf{r}_l \mathbf{r}_l^\top \right) \right) \tag{A.50}$$

$$= I_{2\times 2} \otimes \underbrace{\left(\sum_{l=1}^{2} \Psi'_l \cdot \mathbf{r}_l \mathbf{r}_l^\top \right)}_{=: A \in \mathbb{R}^{2\times 2}} \tag{A.51}$$

$$= \begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix} . \tag{A.52}$$

B Parameter Settings

B.1 Variational 3D Reconstruction

In case of our 3D reconstruction method the solver related parameters mainly effect the trade off between the reconstruction quality and the runtime performance and thus have been fixed in advance, see Table B.1. In contrast, the model parameters have a direct impact on the quality and thus must be selected more carefully. Moreover, from our experience good model parameters may vary depending on the intrinsic camera parameters as well as the distance of the scene to the camera. Nevertheless, even some of the model parameters turned out to be suitable for a wider range of images and thus have also been set fixed in advance, see Table B.1. The remaining weighting parameters of the different terms of the differential energy have been set as specified in Table B.2.

Table B.1: Parameters for the 3D reconstruction method that have been set fixed.

solver parameters		iterations per resolution level	1
		non-linear fixed-point iterations	2
		SOR solver iterations	20
	η	downsampling factor (coarse-to-fine scheme)	0.8
		over-relaxation parameter	1.8
model parameters	ζ	stereo term normalization	0.01
	ϵ	parameter of the regularized linear penalizer Ψ_r	0.001
	ϵ	parameter of the Charbonnier penalizer Ψ_c	0.01
	ϵ	parameter of the Perona-Malik penalizer Ψ_p	0.01

Table B.2: Parameters for the 3D reconstruction methods that have been altered for the different data sets. (*: The parameter ϵ only refers to the penalizer Ψ_1 of the depth regularizer.)

model	data set	ν	α_z	$\alpha_{\mathbf{u}}$	$\alpha_{\mathbf{l}}$	$\alpha_{\boldsymbol{\rho}}$	α_{dz}	$\alpha_{\mathbf{dl}}$	$\alpha_{\mathbf{d}\boldsymbol{\rho}}$	$\epsilon^\star$
combined approach	Blunderbuss Pete	0.1	100	0.1	50	300	0	0.5	0.5	0.001
	Angel	3.0	200	0.1	10	40	0	3.0	3.0	0.005
	Fountain	1.0	290	0.1	10	390	0	0.5	0.5	0.001
	Herz-Jesu	1.0	400	0.1	11	395	0	0.5	0.5	0.001
pure stereo approach	Blunderbuss Pete	–	100	0.1	–	–	0	–	–	0.005
	Angel	–	400	0.1	–	–	0	–	–	0.005
	Fountain	–	300	0.1	–	–	0	–	–	0.001
	Herz-Jesu	–	400	0.1	–	–	0	–	–	0.001

B.2 Variational Motion Estimation

B.2.1 Comparison of Second-Order Regularizers

In case of the different motion estimation methods presented in Section 4.1 we also set most parameters fixed, see Table B.5. The remaining weighting parameters α, β, γ and λ were optimized w.r.t. the average endpoint error on a small subset of the training data sets of the respective benchmarks [14] and are specified in Table B.3.

Table B.3: Parameters for the motion estimation methods of Section 4.1 that have been adjusted.

benchmark	model			γ	α	λ	β
KITTI 2012	direct	first-order	isotropic	5.76	8.78	–	0.81
	direct	first-order	anisotropic	5.37	5.58	–	1.08
	direct	second-order	isotropic	5.22	21.61	–	1.33
	direct	second-order	single anisotropic	6.00	2.90	–	1.49
	direct	second-order	double anisotropic	6.06	3.01	–	1.45
	inf-conv.	second-order	isotropic	7.62	8.93	1.35	1.06
	inf-conv.	second-order	single anisotropic	8.10	9.18	1.17	0.55
	inf-conv.	second-order	double anisotropic	8.83	9.47	0.61	0.43
	coupling	second-order	isotropic	4.19	5.30	29.62	0.53
	coupling	second-order	single anisotropic	5.43	5.29	32.66	0.65
	coupling	second-order	double anisotropic	5.43	5.24	32.65	0.65
KITTI 2015	direct	first-order	isotropic	6.10	7.56	–	0.83
	direct	first-order	anisotropic	5.85	4.83	–	0.58
	direct	second-order	isotropic	6.11	5.99	–	0.72
	direct	second-order	single anisotropic	6.15	2.10	–	0.62
	direct	second-order	double anisotropic	6.20	2.07	–	0.77
	inf-conv.	second-order	isotropic	13.09	13.09	0.62	0.52
	inf-conv.	second-order	single anisotropic	12.68	12.27	0.29	0.81
	inf-conv.	second-order	double anisotropic	14.24	12.99	0.31	0.56
	coupling	second-order	isotropic	3.90	3.64	24.89	0.61
	coupling	second-order	single anisotropic	6.90	5.03	16.14	0.64
	coupling	second-order	double anisotropic	6.80	5.67	12.15	0.76

B.2.2 An Order-Adaptive Regularization Strategy

Regarding the order-adaptive regularization presented in Section 4.2 we again used a fixed set of parameters for the minimization scheme as well as for some model parameters, see Table B.6. The remaining parameters γ, α, λ, θ, τ and κ were set individually for each benchmark as listed in Table B.4.

Table B.4: Parameters for the motion estimation methods of Section 4.2 that have been optimized.

data set/model		γ	α	λ	θ	τ	κ
classroom sequences							
first-order		20.74	61.28	–	–	–	–
second-order		17.66	82.39	58.80	–	–	–
adaptive order	global	17.79	80.02	54.16	$1.0 \cdot 10^{-5}$	$2.0 \cdot 10^{-1}$	–
adaptive order	local	7.13	17.51	49.87	$1.1 \cdot 10^{-4}$	$7.2 \cdot 10^{-4}$	–
adaptive order	non-local	4.03	16.65	39.33	$1.0 \cdot 10^{-5}$	$2.5 \cdot 10^{-4}$	–
adaptive order	region	12.13	57.13	54.61	$8.2 \cdot 10^{-4}$	$7.6 \cdot 10^{-5}$	$1.2 \cdot 10^{-3}$
Middlebury benchmark							
first-order		5.00	18.50	–	–	–	–
second-order		6.00	21.67	203.24	–	–	–
adaptive order	global	4.30	10.27	19.55	$5.5 \cdot 10^{-3}$	$1.3 \cdot 10^{-5}$	–
adaptive order	local	4.16	9.95	19.12	$1.4 \cdot 10^{-2}$	$7.6 \cdot 10^{-3}$	–
adaptive order	non-local	4.29	10.22	19.07	$9.3 \cdot 10^{-3}$	$6.5 \cdot 10^{-3}$	–
adaptive order	region	8.53	15.52	26.71	$1.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-3}$	$3.9 \cdot 10^{-3}$
KITTI 2012 benchmark							
first-order		60.00	145.00	–	–	–	–
second-order		54.94	166.85	17.00	–	–	–
adaptive order	global	42.16	118.59	17.59	$1.7 \cdot 10^{-4}$	$7.4 \cdot 10^{-4}$	–
adaptive order	local	44.24	111.10	26.53	$1.8 \cdot 10^{-4}$	$2.6 \cdot 10^{-6}$	–
adaptive order	non-local	46.80	135.33	13.86	$6.7 \cdot 10^{-6}$	$6.9 \cdot 10^{-6}$	–
adaptive order	region	56.78	150.47	19.27	$1.0 \cdot 10^{-4}$	$4.0 \cdot 10^{-5}$	$1.7 \cdot 10^{-3}$
KITTI 2015 benchmark							
first-order	–	72.50	128.75	–	–	–	–
second-order	–	69.35	186.21	9.84	–	–	–
adaptive order	global	50.72	117.21	9.09	$1.0 \cdot 10^{-4}$	$3.5 \cdot 10^{-4}$	–
adaptive order	local	64.61	120.97	8.61	$2.7 \cdot 10^{-4}$	$1.5 \cdot 10^{-4}$	–
adaptive order	non-local	55.58	152.42	4.66	$1.1 \cdot 10^{-4}$	$2.8 \cdot 10^{-4}$	–
adaptive order	region	41.46	110.86	6.98	$5.0 \cdot 10^{-4}$	$2.4 \cdot 10^{-4}$	$9.4 \cdot 10^{-3}$
MPI Sintel benchmark							
first order		45.00	152.50	–	–	–	–
second order		4.61	39.35	97.25	–	–	–
adaptive order	global	16.49	38.85	176.93	$1.0 \cdot 10^{-6}$	$5.0 \cdot 10^{-5}$	–
adaptive order	local	12.84	29.25	8.77	$2.8 \cdot 10^{-2}$	$5.1 \cdot 10^{-2}$	–
adaptive order	non-local	12.70	29.12	44.41	$2.1 \cdot 10^{-2}$	$3.3 \cdot 10^{-3}$	–
adaptive order	region	36.39	63.82	40.08	$3.5 \cdot 10^{-5}$	$1.5 \cdot 10^{-2}$	$8.0 \cdot 10^{-3}$

Table B.5: Parameters for the motion estimation methods of Section 4.1 that have been set fixed.

solver parameters		iterations per resolution level	1
		number of cascades	max. 10
		non-linear fixed-point iterations	3
		SOR solver iterations	20
	η	downsampling factor coarse-to-fine scheme	0.95
		over relaxation parameter	1.85
model parameters	ζ	data term normalization parameter	0.01
	ϵ	parameter of the penalizer functions Ψ	0.01

Table B.6: Parameters for the motion estimation methods of Section 4.2 that have been set fixed.

solver parameters		iterations per resolution level	1
		number of cascades	max. 10
		non-linear fixed-point iterations	5
		SOR solver iterations	20
	η	downsampling factor coarse-to-fine scheme	0.95
		over relaxation parameter	1.85
(explicit scheme)		iterations	100
		step-size	100
model parameters	ζ	data term normalization parameter	0.01
	ϵ	parameter of the penalizer functions Ψ	0.01

B.3 Beyond Variational Motion Estimation

In case of the order-adaptive illumination-aware refinement model we also used a fixed set of parameters for the minimization scheme as well as for some model parameters, see Table B.7. The remaining parameters α, λ, β were set individually for each benchmark as listed in Table B.8.

Table B.7: Parameters for the motion estimation methods of Chapter 5 that have been set fixed.

solver parameters		number of resolution levels	10
		iterations per resolution level	1
		number of cascades	max. 10
		non-linear fixed-point iterations	5
		SOR solver iterations	20
	η	downsampling factor coarse-to-fine scheme	0.95
		over relaxation parameter	1.85
model parameters	ζ	data term normalization parameter	0.01
	ϵ	parameter of the penalizer functions Ψ	0.01
	γ	gradient constancy weight	5
	τ	minimum average benefit	10^{-5}
	θ	weight/slope factor selection term	10^{-5}

Table B.8: Parameters for the motion estimation methods of Chapter 5 that have been optimized.

	KITTI 2012			KITTI 2015			Sintel		
	α	λ	β	α	λ	β	α	λ	β
our refinement ($\eta = 1.00$)									
DeepMatches	17.92	7.85	0.68	16.78	13.23	0.43	14.86	22.33	0.30
DiscreteFlow	15.11	11.30	0.40	15.00	10.00	0.40	13.50	20.50	0.25
CPM	8.26	8.01	0.44	14.85	7.38	0.53	13.50	20.50	0.25
our refinement ($\eta = 0.95$)									
DeepMatches	13.61	9.67	0.60	18.27	9.75	0.52	12.94	23.45	0.28
DiscreteFlow	12.04	10.19	0.55	15.12	7.92	0.53	13.50	20.50	0.25
CPM	9.98	9.86	0.45	11.73	7.17	0.14	13.50	20.50	0.25
our refinement ($\eta = 0.90$)									
DeepMatches	10.97	5.83	0.46	11.71	6.20	0.47	4.56	181.13	0.24
DiscreteFlow	8.80	7.32	0.28	11.87	6.67	0.32	13.86	21.65	0.20
CPM	8.12	6.64	0.41	10.51	5.35	0.11	13.03	21.43	0.26

B.4 Multi-Frame Motion Estimation

As in all previous cases the variational refinement employed in Section 6.1 and Section 6.2 uses a fixed set of parameters regarding the minimization scheme, see Table B.9. The remaining weighting parameters either have been set fixed per benchmark, see Table B.10 (individual setting), or have been set fixed across all benchmarks, see Table B.10 (single setting). While the individual parameter setting was chosen by optimizing [14] the baseline, as specified in Section 6.1.6, w.r.t. the respective benchmarks, the single parameter setting was chosen by optimizing the baseline w.r.t. a mixed subset of the training data of all the benchmarks considered in the robust vision challenge (Middlebury, KITTI 2015, MPI Sintel, HD1k).

Table B.9: Parameters for the motion estimation methods of Chapter 6 that have been set fixed.

solver parameters		number of resolution levels	5
		iterations per resolution level	2
		non-linear fixed-point iterations	3
		SOR solver iterations	10
	η	downsampling factor coarse-to-fine scheme	0.95
		over relaxation parameter	1.9
model parameters	ζ	data term normalization parameter	0.01
	ϵ	parameter of the penalizer functions Ψ	0.01
	γ	gradient constancy weight	5
	τ	minimum average benefit	10^{-3}
	θ	weight/slope factor selection term	10^{-5}

Table B.10: Parameters for the motion estimation methods of Chapter 6 that have been optimized.

	KITTI 2012			KITTI 2015			Sintel		
	α	λ	β	α	λ	β	α	λ	β
individual setting	13.03	26.12	0.1	15.00	15.00	0.05	25	60	0.5
single setting	17.5	26.25	0.5	17.5	26.25	0.5	17.5	26.25	0.5

Own Publications

1. Y. C. Ju, D. Maurer, M. Breuß, and A. Bruhn. "Direct variational perspective shape from shading with Cartesian depth parametrisation". *Perspectives in Shape Analysis, Mathematics and Visualization*, 2016, pp. 43–72.
2. D. Maurer and A. Bruhn. "ProFlow: Learning to predict optical flow". In: *Proc. British Machine Vision Conference*. 2018, 1–13. *CVPR 2018 Robust Vison Challenge Runner–Up Award*.
3. D. Maurer, Y. C. Ju, M. Breuß, and A. Bruhn. "An efficient linearisation approach for variational perspective shape from shading". In: *Proc. German Conference on Pattern Recognition*. 2015, pp. 249–261.
4. D. Maurer, Y. C. Ju, M. Breuß, and A. Bruhn. "Combining shape from shading and stereo: A joint variational method for estimating depth, illumination and albedo". *International Journal of Computer Vision* 126:12, 2018, 1342–1366. *Invited Paper*.
5. D. Maurer, Y. C. Ju, M. Breuß, and A. Bruhn. "Combining shape from shading and stereo: A variational approach for the joint estimation of depth, illumination and albedo". In: *Proc. British Machine Vision Conference*. 2016, 76:1–76:14.
6. D. Maurer, N. Marniok, B. Goldlücke, and A. Bruhn. "Structure-from-motion-aware PatchMatch for adaptive optical flow estimation". In: *Proc. European Conference on Computer Vision*. 2018, pp. 575–592.
7. D. Maurer, M. Stoll, and A. Bruhn. "Directional priors for multi-frame optical flow". In: *Proc. British Machine Vision Conference*. 2018, pp. 1–13.
8. D. Maurer, M. Stoll, and A. Bruhn. "Order-adaptive and illumination-aware variational optical flow refinement". In: *Proc. British Machine Vision Conference*. 2017, 662:1–662:13.
9. D. Maurer, M. Stoll, and A. Bruhn. "Order-adaptive regularisation for variational optical flow: Global, local and in between". In: *Proc. International Conference on Scale Space and Variational Methods in Computer Vision*. 2017, pp. 550–562.
10. D. Maurer, M. Stoll, S. Volz, P. Gairing, and A. Bruhn. "A comparison of isotropic and anisotropic second order regularisers for optical flow". In: *Proc. International Conference on Scale Space and Variational Methods in Computer Vision*. 2017, pp. 537–549.
11. H. Men, H. Lin, V. Hosu, D. Maurer, A. Bruhn, and D. Saupe. "Visual quality assessment for motion compensated frame interpolation". In: *International Conference on Quality of Multimedia Experience*. 2019, 1–6. *Best Student Paper Award Candidate*.
12. M. Stoll, D. Maurer, and A. Bruhn. "Variational large displacement optical flow without feature matches". In: *Proc. International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*. 2017, pp. 79–92.

13. M. Stoll, D. Maurer, S. Volz, and A. Bruhn. "Illumination-aware large displacement optical flow". In: *Proc. International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*. 2017, pp. 139–154.

14. M. Stoll, S. Volz, D. Maurer, and A. Bruhn. "A time-efficient optimisation framework for parameters of optical flow methods". In: *Proc. Scandinavian Conference on Image Analysis*. 2017, pp. 41–53.

Bibliography

15. M. Abadi et al. *TensorFlow: Large-scale machine learning on heterogeneous systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.
16. L. Adams and J. Ortega. "A multi-color SOR method for parallel computation". In: *Proc. International Conference on Parallel Processing*. 1982, pp. 53–56.
17. A. Ahmadi and I. Patras. "Unsupervised convolutional neural networks for motion estimation". In: *Proc. IEEE International Conference on Image Processing*. 2016.
18. T. Amiaz and N. Kiryati. "Piecewise-smooth dense optical flow via level sets". *International Journal of Computer Vision* 68:2, 2006, pp. 111–124.
19. M. Bai, W. Luo, K. Kundu, and R. Urtasun. "Exploiting semantic information and deep matching for optical flow". In: *Proc. European Conference on Computer Vision*. 2016, pp. 154–170.
20. C. Bailer, B. Taetz, and D. Stricker. "Flow Fields: Dense correspondence fields for highly accurate large displacement optical flow estimation". In: *Proc. IEEE International Conference on Computer Vision*. 2015, pp. 4015–4023.
21. C. Bailer, B. Taetz, and D. Stricker. "Flow Fields: dense correspondence fields for highly accurate large displacement optical flow estimation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
22. C. Bailer, K. Varanasi, and D. Stricker. "CNN-based patch matching for optical flow with thresholded Hinge embedding loss". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2710–2719.
23. S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. "A database and evaluation methodology for optical flow". *International Journal of Computer Vision* 92:1, 2011, pp. 1–31.
24. L. Bao, Q. Yang, and H. Jin. "Fast edge-preserving PatchMatch for large displacement optical flow". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1510–1517.
25. C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. "PatchMatch: A randomized correspondence algorithm for structural image editing". *ACM Transactions on Graphics* 28:3, 2009, p. 24.
26. T. Basha, Y. Moses, and N. Kiryati. "Multi-view scene flow estimation: A view centered variational approach". *International Journal of Computer Vision* 101:1, 2012, pp. 6–21.
27. M. Bertero, T. A. Poggio, and V. Torre. "Ill-posed problems in early vision". *Proceedings of the IEEE* 76:8, 1988, pp. 869–889.

28. M. J. Black and P. Anandan. "Robust dynamic motion estimation over time". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 1991, pp. 296–302.
29. A. Blake, A. Zisserman, and G. Knowles. "Surface descriptions from stereo and shading". *Image and Vision Computing* 3:4, 1985, pp. 183–191.
30. Blender Foundation. *Blender*. URL: https://blender.org.
31. M. Bleyer, C. Rhemann, and C. Rother. "PatchMatch stereo - Stereo matching with slanted support windows". In: *Proc. British Machine Vision Conference*. 2011, 14:1–14:11.
32. F. A. Bornemann and P. Deuflhard. "The cascadic multigrid method for elliptic problems". *Numerische Mathematik* 75:2, 1996, pp. 135–152.
33. J. Braux-Zin, R. Dupont, and A. Bartoli. "A general dense image matching framework combining direct and feature-based costs". In: *Proc. International Conference on Computer Vision*. 2013, pp. 185–192.
34. K. Bredies, K. Kunisch, and T. Pock. "Total Generalized Variation". *SIAM Journal on Imaging Sciences* 3:3, 2010, pp. 492–526.
35. M. Breuß, E. Cristiani, J.-D. Dorou, M. Falcone, and O Vogel. "Perspective shape from shading: Ambiguity analysis and numerical approximations". *SIAM Journal on Imaging Science* 5:1, 2012, pp. 311–342.
36. 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*. 2004, pp. 25–36.
37. T. Brox and J. Malik. "Large displacement optical flow: descriptor matching in variational motion estimation". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33:3, 2011, pp. 500–513.
38. A. Bruhn and J. Weickert. "Towards ultimate motion estimation: combining highest accuracy with real-time performance". In: *Proc. IEEE International Conference on Computer Vision*. 2005, pp. 749–755.
39. A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnörr. "Variational optical flow computation in real time". *IEEE Transactions on Image Processing* 14:5, 2005, pp. 608–615.
40. A. Bruhn. "Variational optic flow computation: accurate modelling and efficient numerics". PhD thesis. Saarland University, 2006.
41. A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr. "A multigrid platform for real-time motion computation with discontinuity-preserving variational methods". *International Journal of Computer Vision* 70:3, 2006, pp. 257–277.
42. A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr. "Discontinuity-preserving computation of variational optic flow in real-time". In: *International Conference on Scale-Space Theories in Computer Vision*. 2005, pp. 279–290.
43. A. Bruhn, J. Weickert, and C. Schnörr. "Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods". *International Journal of Computer Vision* 61:3, 2005, pp. 211–231.

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*. 2012, pp. 611–625.

45. A. Chambolle and P.-L. Lions. "Image recovery via total variation minimization and related problems". *Numerische Mathematik* 76:2, 1997, pp. 167–188.

46. A. Chambolle and T. Pock. "A first-order primal-dual algorithm for convex problems with applications to imaging". *Journal of Mathematical Imaging and Vision* 40:1, 2011, pp. 120–145.

47. T. Chan, A. Marquina, and P. Mulet. "High-order total variation-based image restoration". *SIAM Journal on Scientific Computing* 22:2, 2000, pp. 503–516.

48. P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. "Deterministic edge-preserving regularization in computed imaging". *IEEE Trans. on Image Processing* 6:2, 1997, pp. 298–311.

49. Q. Chen and V. Koltun. "A simple model for intrinsic image decomposition with depth cues". In: *Proc. IEEE International Conference on Computer Vision*. 2013, pp. 241–248.

50. Q. Chen and V. Koltun. "Full Flow: Optical flow estimation by global optimization over regular grids". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4706–4714.

51. D. Cremers and S. Soatto. "Motion competition: A variational approach to piecewise parametric motion segmentation". *International Journal of Computer Vision* 62:3, 2005, pp. 249–265.

52. J. E. Cryer, P.-S. Tsai, and M. Shah. "Integration of shape from shading and stereo". *Pattern Recognition* 28:7, 1995, pp. 1033–1043.

53. O. Demetz, D. Hafner, and J. Weickert. "The Complete Rank Transform: A tool for accurate and morphologically invariant matching of structures". In: *Proc. British Machine Vision Conference*. 2013.

54. 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*. 2014, pp. 455–471.

55. O. Demetz. "Feature Invariance versus Change Estimation in Variational Motion Estimation". PhD thesis. Universität des Saarlandes, 2015.

56. S. Di Zenzo. "A note on the gradient of a multi-image". *Computer Vision, Graphics and Image Processing* 33, 1986, pp. 116–125.

57. A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. "FlowNet: Learning optical flow with convolutional networks". In: *Proc. IEEE International Conference on Computer Vision*. 2015, pp. 2758–2766.

58. D. Ferstl, C. Reinbacher, R. Ranftl, M. Rüther, and H. Bischof. "Image guided depth upsampling using anisotropic total generalized variation". In: *Proc. International Conference on Computer Vision*. 2013, pp. 993–1000.

59. 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.

60. P. Fua and Y. G. Leclerc. "Object-centered surface reconstruction: Combining multi-image stereo and shading". *International Journal of Computer Vision* 16:1, 1995, pp. 35–56.

61. D. Gadot and L. Wolf. "PatchBatch: A batch augmented loss for optical flow". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4236–4245.

62. S. Galliani, K. Lasinger, and K. Schindler. "Massively parallel multiview stereopsis by surface normal diffusion". In: *Proc. IEEE International Conference on Computer Vision*. 2015, pp. 873–881.

63. S. Galliani and K. Schindler. "Just look at the image: Viewpoint-specific surface normal prediction for improved multi-view reconstruction". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 5479–5487.

64. R. Garg, A. Roussos, and L. Agapito. "A variational approach to video registration with subspace constraints". *International Journal of Computer Vision* 104:3, 2013, pp. 286–314.

65. 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*. 2012, pp. 3354–3361.

66. I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Dover, New York, 2000.

67. T. Gerlich and J. Eriksson. "Optical flow for rigid multi-motion scenes". In: *Proc. IEEE International Conference on 3D Vision*. 2016, pp. 212–220.

68. A. S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers, Inc., 1995.

69. G. Graber, J. Balzer, S. Soatto, and T. Pock. "Efficient minimal-surface regularization of perspective depth maps in variational stereo". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 511–520.

70. S. Grewenig, J. Weickert, and A. Bruhn. "From box filtering to fast explicit diffusion". In: *Proc. German Conference on Pattern Recognition*. 2010, pp. 533–542.

71. M. Grossberg and S. Nayar. "Modeling the space of camera response functions". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26:10, 2004, pp. 1272–1282.

72. M. Grossberg and S. Nayar. "What can be known about the radiometric response from images?" In: *Proc. European Conference on Computer Vision*. 2002, pp. 189–205.

73. F. Güney and A. Geiger. "Deep Discrete Flow". In: *Proc. Asian Conference on Pattern Recognition*. 2016.

74. P. Gwosdek, H. Zimmer, S. Grewenig, A. Bruhn, and J. Weickert. "A highly efficient GPU implementation for variational optic flow based on the Euler-Lagrange framework". In: *Proc. European Conference on Computer Vision*. 2010, pp. 372–383.

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

76. D. Hafner, P. Ochs, J. Weickert, M. Reißel, and S. Grewenig. "FSI schemes: Fast semi-iterative solvers for PDEs and optimisation methods". In: *Proc. German Conference on Pattern Recognition*. 2016, pp. 91–102.

77. T. S. F. Haines and R. C. Wilson. "Integrating stereo with shape-from-shading derived orientation information". In: *Proc. British Machine Vision Conference*. 2007, 84:1–84:10.

78. Y. Han, J.-Y. Lee, and I. S. Kweon. "High quality shape from a single RGB-D image under uncalibrated natural illumination". In: *Proc. IEEE International Conference on Computer Vision*. 2013, pp. 1617–1624.

79. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

80. A. Hewer, J. Weickert, T. Scheffer, H. Seibert, and S. Diebels. "Lagrangian strain tensor computation with higher order variational models". In: *Proc. British Machine Vision Conference*. 2013, pp. 129.1–129.10.

81. B. K. P. Horn and B. G. Schunck. "Determining optical flow". *Artificial intelligence* 17:1-3, 1981, pp. 185–203.

82. M. Hornacek, F. Besse, J. Kautz, A. W. Fitzgibbon, and C. Rother. "Highly overparameterized optical flow using PatchMatch belief propagation". In: *Proc. European Conference on Computer Vision*. 2014, pp. 220–234.

83. D. R. Hougen and N. Ahuja. "Adaptive polynomial modelling of the reflectance map for shape estimation from stereo and shading". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 1994, pp. 991–994.

84. Y. Hu, Y. Li, and R. Song. "Robust interpolation of correspondences for large displacement optical flow". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 481–489.

85. Y. Hu, R. Song, and Y. Li. "Efficient coarse-to-fine PatchMatch for large displacement optical flow". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 5704–5712.

86. J. Hur and S. Roth. "Joint optical flow and temporally consistent semantic segmentation". In: *Proc. ECCV Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving*. 2016, pp. 163–177.

87. J. Hur and S. Roth. "MirrorFlow: Exploiting symmetries in joint optical flow and occlusion estimation". In: *Proc. IEEE International Conference on Computer Vision*. 2017, pp. 312–321.

88. E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. "FlowNet 2.0: evolution of optical flow estimation with deep networks". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1647–1655.

89. E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. "Flownet 2.0: Evolution of optical flow estimation with deep networks". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

90. D. S. Immel, M. F. Cohen, and D. P. Greenberg. "A radiosity method for non-diffuse environments". In: *Proc. SIGGRAPH*. Vol. 20. 4. ACM. 1986, pp. 133–142.

91. J. Janai, F. Güney, J. Wulff, M. Black, and A. Geiger. "Slow Flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3597–3607.

92. J. Janai, F. Güney, A. Ranjan, M. Black, and A. Geiger. "Unsupervised learning of multi-frame optical flow with occlusions". In: *Proc. European Conference on Computer Vision*. 2018.

93. H. Jin, D. Cremers, D. Wang, E. Prados, A. Yezzi, and S. Soatto. "3-D reconstruction of shaded objects from multiple images under unknown illumination". *International Journal of Computer Vision* 76:3, 2008, pp. 245–256.

94. Y. C. Ju, A. Bruhn, and M. Breuß. "Variational perspective shape from shading". In: *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer. 2015, pp. 538–550.

95. Y. C. Ju, S. Tozza, M. Breuß, A. Bruhn, and A. Kleefeld. "Generalised perspective shape from shading with Oren-Nayar reflectance". In: *Proc. British Machine Vision Conference*. 2013.

96. J. Kačur, J. Nečas, J. Polák, and J. Souček. "Convergence of a method for solving the magnetostatic field in nonlinear media". *Aplikace matematiky* 13:6, 1968, pp. 456–465.

97. J. T. Kajiya. "The rendering equation". In: *Proc. SIGGRAPH*. Vol. 20. 4. ACM. 1986, pp. 143–150.

98. S. B. Kang, R. Szeliski, and J. Chai. "Handling occlusions in dense multi-view stereo". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2001, pp. 103–110.

99. R. Kennedy and C. Taylor. "Optical flow with geometric occlusion estimation and fusion of multiple frames". In: *Proc. International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*. 2015, pp. 364–477.

100. D. P. Kingma and J. Ba. "ADAM: A method for stochastic optimization". In: *Proc. International Conference on Learning Representations*. 2015, pp. 1–13.

101. T. Kohlberger, C. Schnörr, A. Bruhn, and J. Weickert. "Domain decomposition for variational optical-flow computation". *IEEE Transactions on Image Processing* 14:8, 2005, pp. 1125–1137.

102. T. Kohlberger, C. Schnörr, A. Bruhn, and J. Weickert. "Domain decomposition for parallel variational optical flow computation". In: *Proc. German Conference on Pattern Recognition*. 2003, pp. 196–203.

103. D. Kondermann, R. Nair, K. Honauer, K. Krispin, J. Andrulis, A. Brock, B. Gussefeld, M. Rahimimoghaddam, S. Hofmann, C. Brenner, et al. "The HCI benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving". In: *Proc. IEEE Workshops Conference on Computer Vision and Pattern Recognition*. 2016, pp. 19–28.

104. G. Kuschk and D. Cremers. "Fast and accurate large-scale stereo reconstruction using variational methods". In: *Proc. IEEE International Conference on Computer Vision Workshops*. 2013, pp. 1–8.

105. F. Langguth, K. Sunkavalli, S. Hadap, and M. Goesele. "Shading-aware multi-view stereo". In: *Proc. European Conference on Computer Vision*. 2016, pp. 469–485.

106. Y. G. Leclerc and A. F. Bobick. "The direct computation of height from shading". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 1991, pp. 552–558.

107. S. Lefkimmiatis, J. P. Ward, and M. Unser. "Hessian Schatten-norm regularization for linear inverse problems". *IEEE Trans. on Image Processing* 22:5, 2013, pp. 1873–1888.

108. J. Lellmann, J.-M. Morel, and C. Schönlieb. "Anisotropic third-order regularization for sparse digital elevation models". In: *Proc. International Conference on Scale Space and Variational Methods in Computer Vision*. 2013, pp. 161–173.

109. 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*. 2013, pp. 61–73.

110. Y. Li, D. Min, M. S. Brown, M. N. Do, and J. Lu. "SPM-BP: Sped-up PatchMatch belief propagation for continuous MRFs". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 4006–4014.

111. Y. Li, Y. Hu, R. Song, P. Rao, and Y. Wang. "Coarse-to-fine PatchMatch for dense correspondence". *IEEE Transactions on Circuits and Systems for Video Technology* 28:9, 2018, pp. 2233–2245.

112. C. Liu, J. Yuen, and A. Torralba. "SIFT Flow: Dense correspondence across scenes and its applications". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33:5, 2011, pp. 978–994.

113. Q. Liu-Yin, R. Yu, A. Fitzgibbon, L. Agapito, and C. Russell. "Better together: Joint reasoning for non-rigid 3D reconstruction with specularities and shading". In: *Proc. British Machine Vision Conference*. 2016, 42:1–42:12.

114. D. G. Lowe. "Distinctive image features from scale-invariant keypoints". *International Journal of Computer Vision* 60:2, 2004, pp. 91–110.

115. M. Lysaker, A. Lundervold, and X. C. Tai. "Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time". *IEEE Trans. on Image Processing* 12:12, 2003, pp. 1579–1590.

116. D. Maurer. "Depth-Driven Variational Methods for Stereo Reconstruction". MA thesis. University of Stuttgart, 2014.

117. R. Mecca, Y. Quéau, F. Logothetis, and R. Cipolla. "A single-lobe photometric stereo approach for heterogeneous material". *SIAM Journal on Imaging Science* 9:4, 2016, pp. 1858–1888.

118. S. Meister, J. Hur, and S. Roth. "UnFlow: Unsupervised learning of optical flow with a bidirectional census loss". In: *Proc. AAAI Conference on Artificial Intelligence*. 2018.

119. M. Menze and A. Geiger. "Object scene flow for autonomous vehicles". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3061–3070.

120. M. Menze, C. Heipke, and A. Geiger. "Discrete optimization for optical flow". In: *Proc. German Conference on Pattern Recognition*. 2015, pp. 16–28.

121. Y. Mileva, A. Bruhn, and J. Weickert. "Illumination-robust variational optical flow with photometric invariants". In: *Proc. German Conference on Pattern Recognition*. 2007, pp. 152–162.

122. P. Moulon, P. Monasse, and R. Marlet. "Adaptive structure from motion with a contrario model estimation". In: *Proc. Asian Conference on Computer Vision*. 2012, pp. 257–270.

123. P. Moulon, P. Monasse, R. Marlet, and Others. *OpenMVG. An Open Multiple View Geometry library*. `https://github.com/openMVG/openMVG`.

124. H.-H. Nagel and W. Enkelmann. "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8:5, 1986, pp. 565–593.

125. V. Nair and G. E. Hinton. "Rectified linear units improve restricted boltzmann machines". In: *Proc. International Conference on Machine Learning*. 2010, pp. 807–814.

126. M. Neoral, J. Šochman, and J. Matas. "Continual Occlusions and Optical Flow Estimation". In: *Proc. Asian Conference on Pattern Recognition*. 2018.

127. R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. "DTAM: Dense tracking and mapping in real-time". In: *Proc. IEEE International Conference on Computer Vision*. 2011, pp. 2320–2327.

128. T. Nir, A. M. Bruckstein, and R. Kimmel. "Over-parameterized variational optical flow". *International Journal of Computer Vision* 76:2, 2008, pp. 205–216.

129. L. Oisel, E. Memin, L. Morin, and C. Labit. "Epipolar constrained motion estimation for reconstruction from video sequences." In: *Proc. SPIE*. Vol. 3309. 1998, pp. 460–468.

130. R. Or-El, R. Hershkovitz, A. Wetzler, G. Rosman, A. M. Bruckstein, and R. Kimmel. "Real-time depth refinement for specular objects". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4378–4386.

131. S. Osher and J. A. Sethian. "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations". *Journal of Computational Physics* 79:1, 1988, pp. 12–49.

132. N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. "Highly accurate optic flow computation with theoretically justified warping". *International Journal of Computer Vision* 67:2, 2006, pp. 141–158.

133. P. Pérez, M. Gangnet, and A. Blake. "Poisson image editing". *ACM Transactions on graphics (TOG)* 22:3, 2003, pp. 313–318.

134. P. Perona and J. Malik. "Scale space and edge detection using anisotropic diffusion". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2, 1990, pp. 629–639.

135. Y. Quéau, F. Lauze, and J.-D. Durou. "A L^1-TV algorithm for robust perspective photometric stereo with spatially-varying lightings". In: *Proc. International Conference on Scale Space and Variational Methods in Computer Vision*. 2015, pp. 498–510.

136. R. Ranftl. "Higher-Order Variational Methods for Dense Correspondence Problems". PhD thesis. Austria: Graz University of Technology, 2014.

137. R. Ranftl, K. Bredies, and T. Pock. "Non-local total generalized variation for optical flow estimation". In: *Proc. European Conference on Computer Vision*. 2014, pp. 439–454.

138. R. Ranftl, S. Gehrig, T. Pock, and H. Bischof. "Pushing the limits of stereo using variational stereo estimation". In: *Proc. IEEE Intelligent Vehicles Symposium*. 2012, pp. 401–407.

139. A. Ranjan and M. J. Black. "Optical flow using a spatial pyramid network". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2720–2729.

140. H. Rashwan, M. Mohamed, M. Garcia, B. Mertsching, and D. Puig. "Illumination robust optical flow model based on histogram of oriented gradients". In: *Proc. German Conference on Pattern Recognition*. 2013, pp. 354–363.

141. Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha. "Unsupervised deep learning for optical flow estimation". In: *Proc. AAAI Conference on Artificial Intelligence*. 2017.

142. Z. Ren, O. Gallo, D. Sun, M.-H. Yang, E. B. Sudderth, and J. Kautz. "A fusion approach for multi-frame optical flow estimation". In: *IEEE Winter Conference on Applications of Computer Vision*. 2019.

143. J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. "Epicflow: Edge-preserving interpolation of correspondences for optical flow". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1164–1172.

144. S. Ricco and C. Tomasi. "Dense Lagrangian motion estimation with occlusions". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 1800–1807.

145. L. Robert and R. Deriche. "Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities". In: *Proc. European Conference on Computer Vision*. 1996, pp. 439–451.

146. E. Rouy and A. Tourin. "A viscosity solutions approach to shape-from-shading". *SIAM Journal on Numerical Analysis* 29, 1992, pp. 867–884.

147. L. Rudin, S. Osher, and E. Fatemi. "Nonlinear total variation based noise removal algorithms". *Physica D* 60, 1992, pp. 259–268.

148. D. Samaras, D. Metaxas, P. Fua, and Y. G. Leclerc. "Variable albedo surface reconstruction from stereo and shape from shading". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2000, pp. 480–487.

149. O. Scherzer. "Denoising with higher order derivatives of bounded variation and an application to parameter estimation". *Computing* 60:1, 1998, pp. 1–27.

150. J. L. Schönberger, E. Zheng, M. Pollefeys, and J. M. Frahm. "Pixelwise view selection for unstructured multi-view stereo". In: *Proc. European Conference on Computer Vision*. 2016, pp. 501–518.

151. C. Schroers, D. Hafner, and J. Weickert. "Multiview depth parameterisation with second order regularisation". In: *Proc. International Conference on Scale Space and Variational Methods in Computer Vision*. 2015, pp. 551–562.

152. C. Schroers, H. Zimmer, L. Valgaerts, A. Bruhn, O. Demetz, and J. Weickert. "Anisotropic range image integration". In: *Proc. German Conference on Pattern Recognition*. Springer. 2012, pp. 73–82.

153. R. Schuster, C. Bailer, O. Wasenmüller, and D. Stricker. "Flowfields++: Accurate optical flow correspondences meet robust interpolation". In: *Proc. IEEE International Conference on Image Processing*. IEEE. 2018, pp. 1463–1467.

154. T. Schuster, L. Wolf, and D. Gadot. "Optical flow requires multiple strategies (but only one network)". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017.

155. B. Semerjian. "A new variational framework for multiview surface reconstruction". In: *Proc. European Conference on Computer Vision*. 2014, pp. 719–734.

156. L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black. "Optical flow with semantic segmentation and localized layers". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3889–3898.

157. S. Shen. "Accurate multiple view 3D reconstruction using patch-based stereo for large-scale scenes". *IEEE Trans. on Image Processing* 22:5, 2013, pp. 1901–1914.

158. E. P. Simoncelli, E. H. Adelson, and D. J. Heeger. "Probability distributions of optical flow". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 1991, pp. 310–315.

159. M. Stoll, S. Volz, and A. Bruhn. "Adaptive integration of feature matches into variational optical flow methods". In: *Proc. Asian Conference on Pattern Recognition*. 2013, pp. 1–14.

160. M. Stoll, S. Volz, and A. Bruhn. "Joint trilateral filtering for multiframe optical flow". In: *Proc. IEEE International Conference on Image Processing*. IEEE. 2013, pp. 3845–3849.

161. C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. "On benchmarking camera calibration and multi-view stereo for high resolution imagery". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2008, pp. 1–8.

162. D. Sun, S. Roth, J. P. Lewis, and M. J. Black. "Learning optical flow". In: *Proc. European Conference on Computer Vision*. 2009, pp. 83–97.

163. D. Sun, X. Yang, M. Y. Liu, and J. Kautz. "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

164. D. Sun, S. Roth, and M. J. Black. "A quantitative analysis of current practices in optical flow estimation and the principles behind them". *International Journal of Computer Vision* 106:2, 2014, pp. 115–137.

165. D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. "Models matter, so does training: An empirical study of CNNs for optical flow estimation". *arXiv preprint arXiv:1809.05571*, 2018.

166. E. Tola, V. Lepetit, and P. Fua. "Daisy: An efficient dense descriptor applied to wide-baseline stereo". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32:5, 2010, pp. 815–830.

167. 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*. 2008, pp. 396–405.

168. Z. Tu, R. Poppe, and R. C. Veltkamp. "Weighted local intensity fusion method for variational optical flow estimation". *Pattern Recognition* 50, 2016, pp. 223–232.

169. L. Valgaerts, A. Bruhn, M. Mainberger, and J. Weickert. "Dense versus sparse approaches for estimating the fundamental matrix". *International Journal of Computer Vision* 96:2, 2012, pp. 212–234.

170. L. Valgaerts, A. Bruhn, and J. Weickert. "A variational model for the joint recovery of the fundamental matrix and the optical flow". In: *Proc. German Conference on Pattern Recognition*. 2008, pp. 314–324.

171. L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, and C. Theobalt. "Lightweight binocular facial performance capture under uncontrolled lighting". *ACM Transactions on Graphics* 31:6, 2012, pp. 1–11.

172. L. Valgaerts, A. Bruhn, H. Zimmer, J. Weickert, C. Stoll, and C. Theobalt. "Joint estimation of motion, structure and geometry from stereo sequences". In: *Proc. European Conference on Computer Vision*. 2010, pp. 568–581.

173. S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. "Three-dimensional scene flow". In: *Proc. IEEE International Conference on Computer Vision*. 1999, pp. 722–729.

174. A. Verri and T. Poggio. "Motion field and optical flow: Qualitative properties". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11:5, 1989, pp. 490–498.

175. C. Vogel, S. Roth, and K. Schindler. "An evaluation of data costs for optical flow". In: *Proc. German Conference on Pattern Recognition*. 2013, pp. 343–353.

176. C. Vogel, K. Schindler, and S. Roth. "3D scene flow estimation with a piecewise rigid scene model". *International Journal of Computer Vision* 115:1, 2015, pp. 1–28.

177. O. Vogel, A. Bruhn, J. Weickert, and S. Didas. "Direct shape-from-shading with adaptive higher order regularisation". In: *Proc. International Conference on Scale Space and Variational Methods in Computer Vision*. 2007, pp. 871–882.

178. O. Vogel, M. Breuß, and J. Weickert. "Perspective shape from shading with non-Lambertian reflectance". In: *Proc. German Conference on Pattern Recognition*. Springer. 2008, pp. 517–526.

179. S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer. "Modeling temporal coherence for optical flow". In: *Proc. International Conference on Computer Vision*. 2011, pp. 1116–1123.

180. S. Wang, S. Ryan Fanello, C. Rhemann, S. Izadi, and P. Kohli. "The global patch collider". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 127–135.

181. A. S. Wannenwetsch, M. Keuper, and S. Roth. "Probflow: Joint optical flow and uncertainty estimation". In: *Proc. IEEE International Conference on Computer Vision*. 2017, pp. 1173–1182.

182. A. Wedel, D. Cremers, T. Pock, and H. Bischof. "Structure-and motion-adaptive regularization for high accuracy optic flow". In: *Proc. IEEE International Conference on Computer Vision*. 2009, pp. 1663–1668.

183. A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers. "Efficient dense scene flow from sparse or dense stereo data". In: *Proc. European Conference on Computer Vision*. Springer. 2008, pp. 739–751.

184. D. Wei, C. Liu, and W. Freeman. "A data-driven regularization model for stereo and flow". In: *3DTV-Conference*. IEEE. 2014.

185. J. Weickert and C. Schnörr. "A theoretical framework for convex regularizers in PDE-based computation of image motion". *International Journal of Computer Vision* 45:3, 2001, pp. 245–264.

186. J. Weickert, M. Welk, and M. Wickert. "L^2-stable nonstandard finite differences for anisotropic diffusion". In: *Proc. International Conference on Scale Space and Variational Methods in Computer Vision*. 2013, pp. 380–391.

187. J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner Stuttgart, 1998.

188. J. Weickert, A. Bruhn, T. Brox, and N. Papenberg. "A survey on variational optic flow methods for small displacements". In: *Mathematical models for registration and applications to medical imaging*. Springer, 2006, pp. 103–136.

189. J. Weickert, A. Bruhn, N. Papenberg, and T. Brox. "Variational optic flow computation: From continuous models to algorithms". *IWCVIA* 3, 2003, pp. 1–6.

190. J. Weickert, S. Grewenig, C. Schroers, and A. Bruhn. "Cyclic schemes for PDE-based image analysis". *International Journal of Computer Vision* 118:3, 2016, pp. 275–299.

191. P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. "DeepFlow: Large displacement optical flow with deep matching". In: *Proc. IEEE International Conference on Computer Vision*. 2013, pp. 1385–1392.

192. M. Werlberger, T. Pock, and H. Bischof. "Motion estimation with non-local total variation regularization". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2010, pp. 2464–2471.

193. M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. "Anistropic Huber-L1 optical flow". In: *Proc. British Machine Vision Conference*. 2009, pp. 1–11.

194. C. Wu, B. Wilburn, Y. Matsushita, and C. Theobalt. "High-quality shape from multi-view stereo and shading under general illumination". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2011, pp. 969–976.

195. C. Wu, M. Zollhöfer, M. Nießner, M. Stamminger, S. Izadi, and C. Theobalt. "Real-time shading-based refinement for consumer depth cameras". *ACM Transactions on Graphics* 33:6, 2014, 200:1–200:10.

196. J. Wulff, L. Sevilla-Lara, and M. J. Black. "Optical flow in mostly rigid scenes". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6911–6920.

197. D. Xu, Q. Duan, J. Zheng, J. Zhang, J. Cai, and T.-J. Cham. "Recovering surface details under general unknown illumination using shading and coarse multi-view stereo". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1526–1533.

198. J. Xu, R. Ranftl, and V. Koltun. "Accurate optical flow via direct cost volume processing". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5807–5815.

199. L. Xu, J. Jia, and Y. Matsushita. "Motion detail preserving optical flow estimation". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34:9, 2012, pp. 1744–1757.

200. K. Yamaguchi, D. McAllester, and R. Urtasun. "Efficient joint segmentation, occlusion labeling, stereo and flow estimation". In: *Proc. European Conference on Computer Vision*. 2014, pp. 756–771.

201. K. Yamaguchi, D. McAllester, and R. Urtasun. "Robust monocular epipolar flow estimation". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 1862–1869.

202. J. Yang and H. Li. "Dense, accurate optical flow estimation with piecewise parametric model". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

203. Y. Yang and S. Soatto. "S2F: Slow-to-fast interpolator flow". In: *Proc. IEEE International Conference on Computer Vision*. 2017, pp. 2087–2096.

204. K.-J. Yoon, E. Prados, and P. Sturm. "Joint estimation of shape and reflectance using multiple images with known illumination conditions". *International Journal of Computer Vision* 86:2-3, 2010, pp. 192–210.

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

206. J. J. Yu, A. W. Harley, and K. G Derpanis. "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness". In: *Proc. Workshops European Conference on Computer Vision*. 2016, pp. 3–10.

207. L. F. Yu, S. K. Yeung, Y. W. Tai, and S. Lin. "Shading-based shape refinement of RGB-D images". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 1415–1422.

208. T. Yu, N. Xu, and N. Ahuja. "Recovering shape and reflectance model of non-Lambertian objects from multiple views". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2004, pp. 226–233.

209. T. Yu, N. Xu, and N. Ahuja. "Shape and view independent reflectance map from multiple views". *International Journal of Computer Vision* 73:2, 2007, pp. 123–138.

210. C. Zach, T. Pock, and H. Bischof. "A duality based approach for realtime TV-L^1 optical flow". In: *Proc. German Conference on Pattern Recognition*. 2007, pp. 214–223.

211. C. Zach, T. Pock, and H. Bischof. "A globally optimal algorithm for robust TV-L^1 range image integration". In: *Proc. IEEE International Conference on Computer Vision*. 2007, pp. 1–8.

212. F. Zhang and B. W. Wah. "Fundamental principles on learning new features for effective dense matching". *IEEE Trans. on Image Processing* 27:2, 2018, pp. 822–836.

213. E. Zheng, E. Dunn, V. Jojic, and J. M. Frahm. "PatchMatch based joint view selection and depthmap estimation". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1510–1517.

214. H. Zhou, B. Ummenhofer, and T. Brox. "DeepTAM: Deep tracking and mapping". In: *Proc. European Conference on Computer Vision*. 2018, pp. 822–838.

215. Y. Zhu, Z. Land, S. Newsam, and A. G. Hauptmann. "Guided optical flow learning". In: *Proc. IEEE Workshops Conference on Computer Vision and Pattern Recognition*. 2017.

216. H. Zimmer, A. Bruhn, and J. Weickert. "Optic flow in harmony". *International Journal of Computer Vision* 93:3, 2011, pp. 368–388.

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

218. M. Zollhöfer, A. Dai, M. Innmann, C. Wu, M. Stamminger, C. Theobalt, and M. Nießner. "Shading-based refinement on volumetric signed distance functions". *ACM Transactions on Graphics* 34:4, 2015, 96:1–96:14.

219. S. Zweig and L. Wolf. "InterpoNet, a brain inspired neural network for optical flow dense interpolation". In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 4563–4572.