Max Planck Institute
for Intelligent Systems
Autonomous Vision Group

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Addressing the Data Scarcity of Learning-based Optical Flow Approaches

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von

Joël Janai

aus Bonn

Tübingen
2020

Tag der mündlichen Qualifikation:	24.04.2020
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Andreas Geiger
2. Berichterstatter:	Prof. Dr. Michael J. Black

Abstract

Learning to solve optical flow in an end-to-end fashion from examples is attractive as deep neural networks allow for learning more complex hierarchical flow representations directly from annotated data. However, training such models requires large datasets, and obtaining ground truth for real images is challenging. Due to the difficulty of capturing dense ground truth, existing optical flow datasets are limited in size and diversity. Therefore, we present two strategies to address this data scarcity problem:

First, we propose an approach to create new real-world datasets by exploiting temporal constraints using a high-speed video camera. We tackle this problem by tracking pixels through densely sampled space-time volumes recorded with a high-speed video camera. Our model exploits the linearity of small motions and reasons about occlusions from multiple frames. Using our technique, we are able to establish accurate reference flow fields outside the laboratory in natural environments. Besides, we show how our predictions can be used to augment the input images with realistic motion blur. We demonstrate the quality of the produced flow fields on synthetic and real-world datasets. Finally, we collect a novel challenging optical flow dataset by applying our technique on data from a high-speed camera and analyze the performance of state of the art in optical flow under various levels of motion blur.

Second, we investigate how to learn sophisticated models from unlabeled data. Unsupervised learning is a promising direction, yet the performance of current unsupervised methods is still limited. In particular, the lack of proper occlusion handling in commonly used data terms constitutes a major source of error. While most optical flow methods process pairs of consecutive frames, more advanced occlusion reasoning can be realized when considering multiple frames. We propose a framework for unsupervised learning of optical flow and occlusions over multiple frames. More specifically, we exploit the minimal configuration of three frames to strengthen the photometric loss and explicitly reason about occlusions. We demonstrate that our multi-frame, occlusion-sensitive formulation outperforms previous unsupervised methods and even produces results on par with some fully supervised methods.

Both directions are essential for future advances in optical flow. While new datasets allow measuring the advancements and comparing novel approaches, unsupervised learning permits the usage of new data sources to train better models.

Kurzfassung

Tiefe neuronale Netze ermöglichen das Erlernen von komplexeren hierarchischen Repräsentationen und machen somit das Ende-zu-Ende Lernen des optischen Flusses attraktiv. Jedoch erfordert das Trainieren solcher Modelle große Datensätzen und die Erzeugung von Grundwahrheiten für reale Bilder ist sehr aufwendig. Aufgrund der Schwierigkeiten dichte Grundwahrheiten zu erfassen, sind existierende Datensätze begrenzt in ihrer Größe und Vielfalt. Wir präsentieren zwei Strategien, um diesen Datenmangel zu lösen:

Zunächst schlagen wir einen Ansatz zur Erstellung neuer realen Datensätze vor, wobei wir mithilfe von Hochgeschwindigkeitskameras strenge zeitliche Annahmen ausnutzen. Wir lösen dieses Problem, indem wir Pixel durch dichte Raum-Zeit-Volumen verfolgen, die mit der Hochgeschwindigkeitskamera aufgenommen wurden. Unser Modell nutzt die Linearität kleiner Bewegungen und schätzt Verdeckungen über mehrere Bilder. Mit unserer Technik sind wir in der Lage, außerhalb des Labors in natürlicher Umgebung genaue Referenzflussfelder zu erzeugen. Außerdem zeigen wir, wie unsere Vorhersagen genutzt werden können, um Bilder mit realistischer Bewegungsunschärfe zu ergänzen. Wir bewerten die Qualität der erzeugten Flussfelder mit synthetischen und realen Datensätzen. Schließlich generieren wir einen neuartigen, herausfordernden optischen Fluss Datensatz, indem wir unsere Methode auf Daten einer Hochgeschwindigkeitskamera anwenden. Wir nutzen diesen Datensatz, um den Stand der Technik im optischen Fluss unter unterschiedlich starker Bewegungsunschärfe zu analysieren.

Außerdem untersuchen wir, wie man aus Daten ohne Grundwahrheiten anspruchsvolle Modelle lernen kann. Unüberwachtes Lernen ist eine vielversprechende Richtung, aber die Leistung der derzeitigen Methoden ist immer noch begrenzt. Insbesondere das Fehlen einer korrekten Handhabung von Verdeckungen in dem gebräuchlichen fotometrischen Vergleich stellt eine große Fehlerquelle dar. Während die meisten optischen Fluss Methoden Paare von aufeinanderfolgenden Einzelbildern verarbeiten, kann eine bessere Schätzung von Verdeckungen realisiert werden, wenn mehrere Einzelbilder betrachtet werden. Wir entwickeln eine Methode für das unüberwachte Lernen von optischem Fluss und Verdeckungen mit mehreren Bildern. Genauer gesagt, nutzen wir die minimale Konfiguration von drei Bildern, um den fotometrischen Vergleich zu verstärken und explizit Verdeckungen zu schätzen. Wir zeigen, dass unsere Formulierung die bestehenden unüberwachten Zwei-Bild-Methoden übertrifft und sogar vergleichbare Ergebnisse mit einigen überwachten Methoden liefert.

Beide Strategien sind für künftige Fortschritte im Bereich des optischen Flusses von wesentlicher Bedeutung. Während neue Datensätze es ermöglichen, die Fortschritte zu messen und neue Ansätze zu vergleichen, erlaubt das unüberwachte Lernen die Nutzung neuer Datenquellen, um bessere Modelle zu trainieren.

Acknowledgments

I would like first to express my sincere gratitude to my advisors Prof. Dr. Andreas Geiger and Prof. Dr. Michael J. Black, for their great support, patience, and motivation. During my Ph.D., Andreas Geiger gave me great guidance in all my research and also for writing this thesis. I am thankful to Michael Black for the excellent research environment we have at the Max-Planck-Institute and the knowledge he shared with me. It was a great honor to work with both Andreas Geiger and Michael Black on the different exciting projects together. In addition, I would like to thank the rest of my committee: Prof. Dr. Carsten Rother and Prof. Dr. Zeynep Akata, for their time and the evaluation of my work.

I am very grateful to all the fantastic people from the Autonomous Vision Group and Perceiving System Department. Special thanks goes to Lars Mescheder, Aseem Behl, Yiyi Liao, Despoina Paschalidou, Michael Niemeyer, Michael Öchsle, Benjamin Coors, Eshed Ohn-Bar, Carolin Schmitt, Simon Donne, Seungjun Nah, Katja Schwarz, Aditya Prakash, Kashyap Chitta, Soubhik Sanyal, Timo Bolkart, Partha Ghosh, Ahmed Osman, Jonas Wulff, Sergey Prokudin, Thomas Nestmeyer, Varun Jampani, Laura Sevilla, Christoph Lassner, Ali Osman Ulusoy, Martin Kiefel, Andreas Lehrmann, and David Stutz, for fruitful discussions and a lot of fun even during stressful times. I am also grateful to the great staff members who always supported us during our research: Kerstin McGaughey, Diana Rebmann, Melanie Feldhofer, Nicole Overbaugh, Telintor Ntounis, Joan Piles, and Jojumon Kavalan. I enjoyed a lot the exciting projects with Anurag Ranjan and I am particularly grateful to Fatma Güney for her help during my Ph.D. and great collaborations.

My sincere thanks also goes to Dr. Richard Szeliski and Prof. Dr. Daniel Scharstein for the great experience and knowledge I gained during my internship at Facebook. I enjoyed working in such an exciting team and spending the summer in such a great city.

Last but not least, I am grateful to my parents and sisters, for the great moral and emotional support throughout my Ph.D. and my life in general. I am also thankful to all my friends who always supported me along the way.

Abbreviations and Symbols

Acronyms

ANN	Approximate Nearest Neighbor
CNN	Convolutional Neural Network
EPE	Average End-point Error
FN	False Negative
FP	False Positive
GPU	Graphical Processing Unit
GT	Ground Truth
HFR	High-Frame-Rate
LiDAR	Light Detection and Ranging
MAP	Maximum A Posteriori
MP-PBP	Max Product Particle Belief Propagation
MRF	Markov Random Field
PBP	Particle Belief Propagation
ReLU	Rectified Linear Unit
RGB	Red, Green and Blue
SOR	Successive Over-Relaxation
TGV	Total Generalized Variation
TN	True Negative
TP	True Positive
TRW-S	Tree-Reweighted Sequential Message Passing
TV	Total Variation

Notation

Scalars	Regular lower (greek) case	$a, b, c, \alpha, \beta, \gamma$
Vectors	Bold lower (greek) case	$\mathbf{a}, \mathbf{b}, \mathbf{c}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$
Matrices	Bold upper case	$\mathbf{A}, \mathbf{B}, \mathbf{C}$
Sets	Calligraphic upper case	$\mathcal{A}, \mathcal{B}, \mathcal{C}$
Functions	Calligraphic lower and upper case	$f(\cdot), g(\cdot), \mathcal{F}(\cdot), \mathcal{G}(\cdot)$

Other symbols:

Number sets	$\mathbb{R}, \mathbb{N}$
Element i of vector $\mathbf{x}$	$\mathbf{x}(i)$
Element $\mathbf{x} = (i, j)^T$ of matrix $\mathbf{A}$	$\mathbf{A}(\mathbf{x}) = \mathbf{A}(i, j)$
Sample i	$x^{(i)}$
Element-wise multiplication	$\cdot \odot \cdot$
Inner product	$\langle \cdot, \cdot \rangle$
L^p Norm	$\lVert \cdot \rVert_p$
Iverson bracket (1 if true, 0 otherwise)	$[\cdot]$
Vector or matrix transpose	$\mathbf{a}^T, \mathbf{A}^T$
Jacobian of matrix $\mathbf{A}$	$\mathbf{J_A}$
Divergence of a vector field $\mathcal{U} = (u(x,y), v(x,y))^T$	$div(\mathcal{U}) = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$
Forward difference in x- and y-direction	∇_x, ∇_y

Fixed symbols:

Energy and loss functions	$\mathcal{E}(\cdot), \mathcal{L}(\cdot)$
Unary and pairwise potentials of a Markov Random Field	ψ^U, ψ^P
Image i	$\mathbf{I}_i$
Continuous, discrete image space	$\mho \subset \mathbb{R}^2, \Omega \subset \mathbb{N}^2$
Optical flow in continuous, discrete image space	$\mathcal{U}, \mathbf{U}$
Occlusion mask in discrete image space	$\mathbf{O}$
Robust penalty function	$\rho(x)$
Fixed parameters	μ, ω, λ
Trainable parameters of a neural network	θ

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

We Humans easily navigate and interact in our 3D world while relying only on 2D observations - projections of the world onto our retina. Nevertheless, we are able to safely perform even complex tasks such as driving a vehicle in urban areas with many traffic participants or on highways with high speed. Thereby, the perception of the motion of objects is an essential cue to obtain a proper understanding of our world. Early psychological studies by Gibson [Gib50; Gib58; Gib66] suggest that humans and animals use optic flow, the change of the retinal image over time, to understand the structure, and be aware of motion in the 3D world. On the one hand, the optic flow of the static scene induced by the change of the observer's location, also called parallax, helps us to perceive distances [Gib50]. When we move around in the world, we can observe how static objects in our proximity move fast while static objects in the far distance move slowly. On the other hand, the optic flow provides information about our motion in the world. A forward motion causes a radial expansion of the optic flow. Shortly before passing (or colliding with) objects, we observe an explosive acceleration of the optic flow. Therefore, flying animals and insects also rely on motion cues for collision avoidance [Gib58], and it has been shown that the time-of-collision can be computed from the optic flow [Lee76].

Optical Flow in Computer Vision: Intelligent systems need a similar understanding of the world to navigate and interact in it. For instance, a vehicle driving autonomously needs to be aware of other traffic participants, detect the road, traffic signs, and traffic lights. Computer vision addresses the perception of intelligent systems and aims to obtain a high-level understanding of the 3D world from images or other sensory input. The optical flow problem was introduced and defined by Horn and Schunck [HS81] in computer vision as the apparent motion of brightness patterns between two consecutive images. One example illustrating the optical flow is provided in Fig. 1.1 with a BMX biker doing a flip. It is common to visualize the optical flow with a color encoding where the hue and saturation represent the direction and magnitude of the motion, respectively. For better illustration, we also overlay the color encoding with sparse directional vectors.

As indicated by the psychological studies, the optical flow is an important cue to address computer vision problems. If we consider the problem of autonomous driving, different traffic participants are likely to show different motion patterns, and the motion of the road, traffic signs and lights are only caused by parallax. An autonomous vehicle also needs to take into account the braking distance for collision avoidance, which is proportional to the driving speed. Towards this goal, it is essential to make accurate predictions of the future trajectory of traffic participants. The observed optical flow can be used to make such predictions into the future and allows a system to take them into consideration when planning the future route.

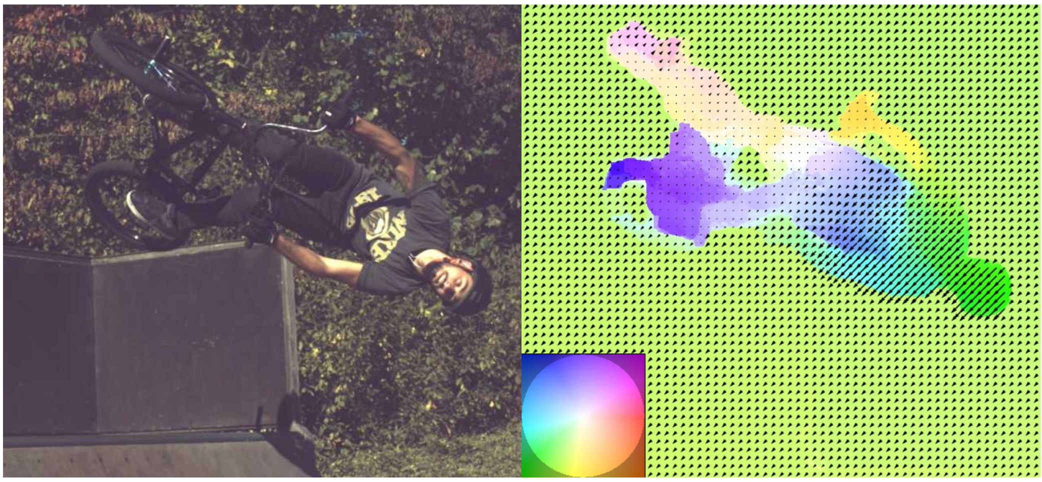


Figure 1.1: Optical Flow. *An example of a challenging sequence of a BMX rider performing a flip. The right image is a visualization of the optical flow with the typical color encoding shown on the bottom left corner and sparse directional vectors. The color encoding uses the hue and saturation to visualize the orientation and magnitude accordingly.*

Optical flow was initially used for video compression and frame interpolation. In both cases, new images are synthesized by mapping image information from the past into the future using optical flow. In the meanwhile, it serves as input for several computer vision tasks dealing with scene analysis. In ego-motion estimation [Gei09; BW16], the optical flow between images from a monocular camera can be used to recover the motion of a system. Furthermore, it has been used to reconstruct the 3D world from images in structure-from-motion pipelines [HJ92; VBW08; WLF14], to track different objects in a scene [Zhu+17b; Zhu+17a; Zhu+18; Wan+18a], and to detect actions performed in short video clips [SZ14; FPZ16].

Estimating Optical Flow: Horn and Schunck [HS81] also proposed the first approach addressing the optical flow problem based on a variational formulation, assuming the brightness of a pixel to be constant over time. While research on this problem has already been carried out for several decades, occlusions, large displacement, and fine details are still challenging for modern methods. One major problem of the optical flow definition is that the motion of the brightness patterns does not necessarily correspond to the motion field, which is the 2D projection of the 3D motion of objects relative to the camera. A good example illustrating this problem was given by Horn and Schunck [HS81] using a uniform sphere. In the case of a rotational motion of the sphere, the apparent motion will be zero because of the uniform appearance, while the motion field will be the 2D projection of the rotational motion. Even in the case of specular reflections, the apparent motion will not reflect the rotational motion of the sphere but the motion of the light source or camera. In tasks such as image interpolation and compression, this ambiguity is not problematic since the task is to replicate brightness patterns. However, the motion field is more relevant in scene analysis since it provides information about the world, e.g., the structure and object motion. Therefore, optical flow is usually used as an approximation of the motion field.

The large degree of ambiguities inherent to the ill-posed optical flow problem can only be resolved using prior knowledge about the appearance and motion of image sequences. Early approaches addressing the optical flow problem [HS81; BA93] integrate simple local smoothness assumptions about the optical flow field using continuous optimization. The introduction of higher-order priors [BKP10; RBP14], patch-based formulations [YMU14; YL15; SSB12] and semantics [Sev+16; Bai+16] allowed the consideration of information over larger image regions to overcome the limitations of local priors.

More recently, deep neural networks [Dos+15; RB16; Ilg+17; Sun+18b] have been successfully applied to the optical flow problem. Learning to solve optical flow in an end-to-end fashion from examples is attractive as deep neural networks allow for learning even better priors from annotated data directly. However, training such models requires large datasets, and obtaining ground truth for real images is challenging. Existing approaches train primarily on synthetic data [Dos+15; May+16], which is cheap to create but does not represent the distribution of real-world scenes. In this work, we discuss two approaches to address this data scarcity problem for optical flow methods. First, we propose a novel approach to generate reference optical flow fields for natural scenes that can be used for evaluation and training of optical flow methods. Second, we consider the problem of learning optical flow from data without any annotations.

Generating Data using High-Speed Cameras: The recent strong progress in other computer vision tasks was mostly driven by high-capacity models (deep neural networks) trained on very large annotated datasets. One prominent example is the advancement in image classification initiated by ImageNet [KSH12; Rus+15]. Other examples are object localization with MS COCO [Lin+14] and semantic segmentation with Cityscapes [Cor+16].

The acquisition of such large annotated datasets for the optical flow problem is complicated. In contrast to other problems like stereo and reconstruction, where active sensors such as structured light or laser scanners can be used, there exists no such sensor to record the optical flow ground truth. Furthermore, a sub-pixel accurate manual annotation is infeasible even with crowdsourcing efforts such as Amazon Mechanical Turk [Ama]. Baker et al. [Bak+11] addressed this problem in a lab setting using fluorescent ink in combination with UV light to track pixels over time. Geiger, Lenz, and Urtasun [GLU12] and Kondermann et al. [Kon+16] use a laser scanner and obtain correspondences by reprojecting the points into the image plane. While this only provides ground truth for the static scene, Menze and Geiger [MG15] also manually annotates dynamic objects such as cars by aligning 3D CAD models. However, these methods do not scale up and can only be applied in certain environments because of the complex setup.

To address these problems, we propose to exploit the power of high-speed video cameras for creating accurate optical flow reference data in a variety of natural scenes. The idea is to leverage the temporal information from high-frame-rate sequences to accurately estimate the optical flow of the corresponding sequences in a standard frame rate. Considering the large space-time volume spanned by the High-Frame-Rate (HFR) sequence directly is difficult because of the large number of unknown parameters. Therefore, we split the problem into two simpler subproblems: First, we estimate the optical flow between intermediate HFR frames, and second, we combine the intermediate optical flow fields to obtain the low-frame-rate optical flow. This can be done simply by estimating the optical flow fields between intermediate HFR frames using a popular optical flow method and summing up the flow fields along the trajectories. However, this approach achieves weak performance since small errors are accumulated to a large drift, and occlusions are not taken into account.

Instead, we develop a multi-frame extension of a classical variational approach to leverage temporal information in HFR sequences. While the majority of optical flow methods only consider two frames for estimation, a multi-frame formulation has the advantage that more observations can be used to resolve ambiguities and improve the estimation in occluded regions. In addition, this formulation allows us to reason about occlusions elegantly and obtain sharper motion boundaries. Finally, we propose a dense tracking method addressing the drift and occlusion estimation problem of the simple accumulation. We make the optimization over the large space-time volume spanned by the HFR sequence feasible by considering the solution of intermediate frames using our multi-frame approach. Our method achieves very accurate motion estimations in visible and occluded regions on a synthetic dataset. Eventually, we use it to benchmark several state-of-the-art optical flow methods and analyze their performance.

Unsupervised Learning of Flow and Occlusions: The problem with supervised learning is the dependency on accurate annotations. The diversity and size of the dataset are important for training sophisticated models and avoiding overfitting. In the case of overfitting, a model is not able to generalize to data following a different distribution as the training data, which might lead to weak performance on new observations. Especially in safety-critical applications such as autonomous driving, the learned models must be reliable and achieve comparable performance as during training. Sophisticated models consisting of a large number of trainable parameters are prone to overfitting since they can memorize characteristics of the training dataset [GBC16].

In contrast, unsupervised learning opens the opportunity to learn optical flow from any data source available, as from large internet video collections. While several approaches have been proposed for learning optical flow in an unsupervised fashion [YHD16; Vij+17; PHC16; Ren+17; Wan+18b; MHR18], none have achieved competitive results with supervised methods. One major reason for the weak performance is the loss used for learning. Inspired by classical methods, the loss enforces the assumption that brightness patterns are constant over time (photometric loss). While this assumption provides good guidance in textured visible regions, it is strongly misleading in occluded regions due to the lack of information.

Only a few approaches [Wan+18b; MHR18] proposed heuristics to estimate these occluded regions and ignore them during learning. We propose a formulation using three frames to jointly learn optical flow and occlusions in an unsupervised fashion. Instead of just masking out occluded regions in the photometric loss, we model the occlusions and provide a training signal to learn them simultaneously. This differentiable occlusion reasoning allows us to train the model end-to-end and obtain better occlusions. Furthermore, our three frame formulation leverages future and past information to improve the predictions in occluded regions. Eventually, our unsupervised method outperforms all previous methods and even achieves comparable results to a few supervised methods.

1.1 Contribution

The contributions of this thesis can be summarized as:

- We create a novel real-world dataset obtained using a high-speed camera:
  - We generate the reference data using an accurate multi-frame optical flow method and long-term tracking formulation.
  - Our multi-frame optical flow method leverages strong temporal constraints to jointly reason over optical flow and occlusions for accurate motion estimation from HFR sequences with sharp motion boundaries.
  - Our long-term tracking formulation leverages the accurate flow estimates obtained with our multi-frame optical flow approach to make the optimization feasible. Our formulation allows us to alleviate the drift problem and handle occlusions.
  - Finally, we provide an evaluation of several state-of-the-art optical flow approaches on our novel dataset. We synthesize motion blur and consider different motion lengths in our benchmark. In our analysis, we identify the strengths and weaknesses of the different approaches.
- We present a scheme for unsupervised learning of optical flow and occlusions:
  - We propose a photometric loss that leverages multiple frames to obtain a training signal for both the optical flow and occlusions. We make our loss differentiable by using continuous occlusion variables. This allows us to train the whole model end-to-end.
  - The photometric loss ignores occlusions by either focusing on past or future information, depending on the occlusion. Misleading gradients are suppressed during training by relying only on available image information.
  - We propose a multi-frame extension of a state-of-the-art network that allows us to learn occlusions. At the same time, we suggest a few changes of the architecture to improve the performance of the network.

1.2 Overview

We start with the mathematical foundations of methods used in this thesis in Chapter 2 and an introduction to the optical flow problem in Chapter 3. We discuss different methods to address the optical flow problem and relate them to the approaches proposed in this thesis. Finally, we review the available datasets for the optical flow problem. We are particularly interested in the questions of how they are created and what their limitations are. Chapter 4 treats the problem of generating optical flow ground truth in natural scenes from HFR sequences. We develop a method to generate reference data, which is accurate enough for the evaluation of other methods and training of deep neural network models. Towards this goal, our formulation leverages strong temporal constraints in HFR sequences to reason over optical flow and occlusions jointly. Using this method, we create a novel real-world dataset and use it to compare state-of-the-art optical flow methods. In Chapter 5, we consider the problem of learning optical flow from data without any annotations. We extend a state-of-the-art architecture to leverage multiple frames for unsupervised learning of optical flow and occlusions. The multi-frame formulation allows us to use past or future information to improve the estimation in occluded regions. We present different losses for this task and show how our formulation outperforms all previous unsupervised methods. Finally, we conclude our work in Chapter 6 and discuss different future directions for both approaches. Besides individual future opportunities, we also see great potential in combining the presented ideas to create larger and more diverse datasets for the optical flow problem.

2 Foundations

In this chapter, we will shortly introduce the mathematical foundations of the methods used in the presented approaches. We will first give a brief introduction into the calculus of variations (Section 2.1), which will be the foundation of our HFR flow estimation method described in Section 4.2. Afterwards in Section 2.2, we will introduce the concept of Markov Random Fields (MRFs) and how inference is performed in such probabilistic models. In Section 4.3, we will derive a simple MRF from our objective function to make optimization feasible. In addition, we will rely on graph cuts to optimize our binary occlusion variables in our HFR flow estimation discussed in Section 4.2. Finally, we introduce Convolutional Neural Networks (CNNs) in Section 2.3 that we will be the foundation of the second part (Chapter 5) on unsupervised learning.

2.1 Calculus of Variations

Several computer vision problems such as optical flow, denoising, deblurring, depth estimation, and 3D reconstruction can be formulated as finding a continuous function mapping image pixels to real values (e.g., flow, intensity, or 3D structure). While the discretization of images needs to be taken into account with some approximations like bilinear interpolations, such continuous formulations have the advantage that well understood mathematical concepts can be applied. The mathematical field calculus of variations [Els12] covers the problem of finding maxima and minima of functionals. A functional $\mathcal{S}$ is a mapping from functions $\mathcal{F}$ with argument $\mathbf{p} \in \mho \subseteq \mathbb{R}^d$ of d dimensions and derivative $\mathcal{F}'$ to real numbers, and is usually defined as integral over functions and their derivatives:

$$\mathcal{S}\left(\mathcal{F}(\mathbf{p})\right) = \int\limits_{\mho} \mathcal{L}(\mathbf{p}, \mathcal{F}(\mathbf{p}), \mathcal{F}'(\mathbf{p})) d\mathbf{p} \tag{2.1}$$

The derivatives of the functions allow incorporating constraints on the functions. Constraining the solution space is beneficial when dealing with inverse problems arising in computer vision. Usually, many solutions can explain the same observations due to the loss of information when capturing images, such as 3D information. However, assumptions based on properties of the real world can be easily enforced on the solution using the calculus of variations. A popular assumption that will be used in this thesis is the smoothness constraint that favors smooth functions.

The extremum of a functional can be obtained by setting the derivative of the functional to zero and solving the second-order partial differential equation, the Euler-Lagrange equation.

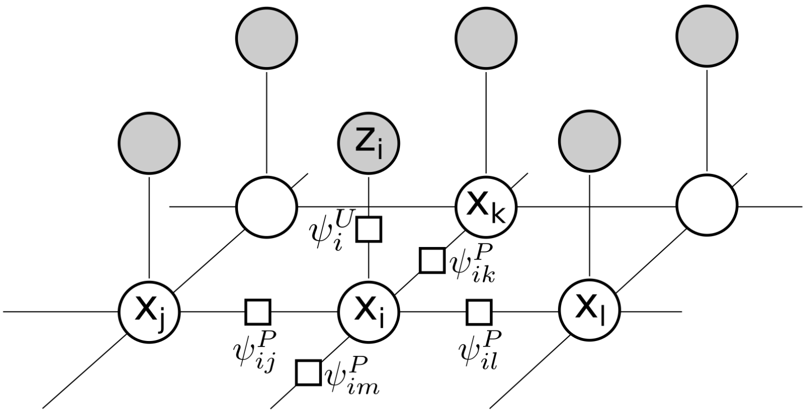


Figure 2.1: **Markov Random Field.** *Illustration of the neighborhood of random variable X_i with unary term ψ_i^U and pairwise terms ψ^P to its four neighbors.*

Thus, a function $\mathcal{F}$ is the extremum of $\mathcal{S}$ if the Euler-Lagrange equation is satisfied:

$$\frac{\partial \mathcal{L}}{\partial \mathcal{F}} + \sum_{i}^{d} \frac{\partial}{\partial p(i)} \cdot \left(\frac{\partial \mathcal{L}}{\partial \mathcal{F}'_i} \right) = \frac{\partial \mathcal{L}}{\partial \mathcal{F}} + div(\frac{\partial \mathcal{L}}{\partial \mathcal{F}'}) = 0 \tag{2.2}$$

with $\mathcal{F}'_i = \frac{\partial \mathcal{F}}{\partial p(i)}$.

Variational optical flow approaches [HS81; LK81; Bro+04] have been very popular due to the elegant incorporation of constraints and their high accuracy, especially for small motion, as discussed in detail in Chapter 3. In optical flow, we can formulate an energy functional $\mathcal{E}$ over flow fields $\mathcal{U}$ consisting of a data term $\mathcal{E}^{\mathcal{D}}$ and a smoothness prior $\mathcal{E}^{\mathcal{S}}$

$$\mathcal{E}(\mathcal{U}) = \int_{\mho} \mathcal{E}^{\mathcal{D}}(\mathcal{U}) + \mathcal{E}^{\mathcal{S}}(\mathcal{U}) \tag{2.3}$$

with $\mho$ the image space. While the data term uses image evidence to validate the flow fields, the smoothness assumption is a constraint on the derivative of flow fields. Eventually, we use the Euler-Lagrange equation to find the flow field $\mathcal{U}$ that minimizes the energy. For the generation of reference data from HFR sequences, we will estimate small motions between intermediate frames in Section 4.2. These estimates should be as accurate as possible since we rely on them in our dense tracking formulation (Section 4.3). Small errors complicate the dense tracking problem and might lead to larger drift. Therefore, we extend a variational approach to estimate the optical flow and occlusions between HFR frames by incorporating strong temporal constraints leading to great improvements in visible and occluded regions.

2.2 Bayesian Formulation

Bayesian formulations [Bar12; Sze11] provide another elegant framework to address computer vision problems. In contrast to the calculus of variations, they simultaneously model the degree of belief and can be easily applied to problems involving discrete variables. Similar to variational approaches, constraints on the solutions can be incorporated using priors. Eventually, similar energies, as in Eq. (2.3), can be derived from probabilistic models. We will rely on such probabilistic models to infer binary occlusions variables in Section 4.2 and to jointly infer continuous and discrete variables in Section 4.3 while enforcing temporal and spatial assumptions.

MRFs are popular probabilistic models for grid-based inference problems. The model is defined over an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{C})$. Each node represents a pixel, which is associated with a random variable, and the edges $\mathcal{C}$ represent the dependency between the random variables. In our case, we are interested in hidden variables X with discrete state space while considering the images as observations Z (Fig. 2.1). Now, we can infer our hidden variables by maximizing the posterior distribution using the Bayes rule

$$P(x|z) = \frac{P(z|x) \cdot P(x)}{P(z)} = \eta P(z|x) \cdot P(x), \tag{2.4}$$

with x denoting $X = x$, z denoting $Z = z$ and a proportional factor η such that $\sum_x P(x|z) = 1$.

Computations in this graph are made feasible using the Markov property, which states that each random variable is conditionally independent, given a subset of random variables (maximal cliques of $\mathcal{G}$). This way, the joint probability is factorized into a product of conditional probabilities over nodes i and pair of cliques $(i, j) \in \mathcal{C}$:

$$P(z|x) = \prod_{i=1}^{N} P(z_i|x_i) = \prod_{i=1}^{N} exp\left(\psi_i^U(x_i)\right) \tag{2.5}$$

$$P(x) = \prod_{(i,j)\in\mathcal{C}} P(x_i|x_j) = \prod_{(i,j)\in\mathcal{C}} exp\left(\psi_{ij}^P(x_i,x_j)\right) \tag{2.6}$$

Thus, we can rewrite the posterior distribution using the unary potential ψ^U and pairwise potential ψ^P

$$P(x|z) = \eta \prod_{i=1}^{N} exp\left(\psi_i^U(x_i)\right) \cdot \prod_{(i,j)\in\mathcal{C}} exp\left(\psi_{ij}^P(x_i,x_j)\right). \tag{2.7}$$

Inference in an MRF is performed using Maximum A Posteriori (MAP) by maximizing the negative log-likelihood, which is equivalent to minimizing the energy $\mathcal{E}_{MRF}(X)$. Since η is constant, we can ignore the proportional factor during minimization.

$$\begin{aligned}\mathcal{E}_{MRF}(X) &= log\left(P\left(x|z\right)\right) \\ &= \sum_{\mathbf{p}} \psi_i^U\left(x_i\right) + \sum_{\mathbf{p}\sim\mathbf{q}} \psi_{ij}^P(x_i,x_j).\end{aligned} \tag{2.8}$$

Note that the equation is similar to Eq. (2.3) when considering a discrete image space. The first term can be considered as data term $\mathcal{E}^{\mathcal{D}}$ and the second term as smoothness constraint $\mathcal{E}^{\mathcal{S}}$. Instead of integrating over the image space, we consider the discretized image space and sum over pixels.

2.2.1 Inference

Different algorithms [Bar12] have been developed to infer hidden variables in such graphical models. A popular optimization method to compute the Maximum A Posteriori (MAP) estimate of MRF is loopy belief propagation. Towards this goal, messages (intermediate results) are passed in forward and backward direction in the graph. While belief propagation provides an exact solution on trees, the cycles (loops) in graphs only allow for an approximate solution. Two variants of belief propagation exist. The max-product algorithm directly returns the maximum in each step while the sum-product algorithm computes marginal distributions of each node. Different algorithms have been proposed to improve the results on graphs. Tree-Reweighted Sequential Message Passing (TRW-S) [Kol06] splits the graph into a set of trees to compute probability distributions over these trees and use them to reweight the messages during belief propagation. In addition, TRW-S computes a lower bound on the energy, which allows comparing the energy of the approximation to the lower bound (global optimum).

Particle Belief Propagation

These inference techniques become computationally infeasible when dealing with continuous or even a combination of continuous and discrete variables. Inspired by the particle filter, Koller, Lerner, and Anguelov [KLA99] suggest to discretize the continuous variables and sample the discrete variables. Thus, each distribution will be represented by a finite set of samples (particles). The set of particles can either be drawn for each message or each variable. Both approaches yield the correct solution when the number of particles goes to infinity. However, drawing the set of particles for each variable has the advantage that they can be considered as possible values of these random variables. Thus, inference reduces to an alternation between finding the MAP of the discrete MRF and resampling particles from the current solution [IM09].

Ihler and McAllester [IM09] suggest initializing the particles using local potentials, but in the case of continuous variables, this is difficult because of the high dimensional space. While random sampling has been used successfully [Bes+14], a data-driven initialization provides better results. In each iteration, the particles can be resampled by drawing from a Gaussian distribution centered at the current solution. Besse et al. [Bes+14] also suggested

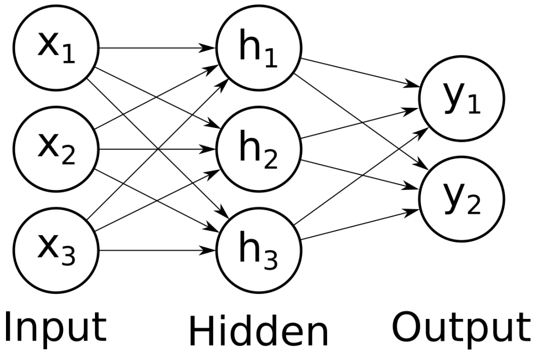


Figure 2.2: Neural Network *Multi-layer neural network with one hidden layer.*

to use the particles from neighbors in each iteration to generate new samples and spatially propagate particles.

Our dense tracking formulation discussed in Section 4.3.3 consists of discrete and continuous variables. We rely on Particle Belief Propagation (PBP) to make optimization feasible by discretizing the continuous variables and sampling the discrete variables. This allows us to derive a simpler MRF that can be optimized using the TRW-S algorithm. We use a data-driven initialization of the particles, and in each iteration, we resample particles from local neighborhoods for spatial propagation.

Graph Cuts

Another popular method to compute the MAP estimate of an MRF with discrete variables by optimizing the energy function in Eq. (2.8) is graph cuts [BVZ99; KZ04]. Especially for binary variables, graph cuts algorithms are attractive since they return the optimal solution [GPS89]. Therefore, we rely on graph cuts to optimize our binary occlusions variables when estimating the optical flow between HFR frames in Section 4.2.4. We alternate between a continuous optimization of the optical flow using a variational approach and a discrete optimization of the occlusions variables. In each iteration given our current optical flow estimates, graph cuts will return the optimal solution of our binary occlusion variables.

For binary variables, a new graph $\mathcal{G}'$ is constructed consisting of nodes $\mathcal{V}' = \{v_0, \dots, v_N, s, t\}$ with $\{v_0, \dots, v_N\}$ representing the pixels and two terminal nodes $\{s, t\}$ (source and sink) representing the binary states (our occlusion states). The edge weights between the pixel and terminal nodes correspond to the unary terms with corresponding binary states, and the edge weights between pixel nodes correspond to the pairwise terms. A minimum cut in G' is a partitioning $\mathcal{S}, \mathcal{T} \subset \mathcal{V}'$ of nodes such that $s \in \mathcal{S}$, $t \in \mathcal{T}$, $\mathcal{S} \cap \mathcal{T} = \emptyset$ while the sum of edge weights going from $\mathcal{S}$ to $\mathcal{T}$ is minimal. Finding the minimum cut on the graph yields the global optimum of the energy function.

2.3 Neural Networks

The discrete and continuous optimization methods discussed so far are computationally expensive. Learning-based approaches instead learn to solve such problems from data directly. While the learning itself is still computationally demanding, the application of learned models is usually much more efficient than classical optimization methods. However, large datasets are necessary to learn such sophisticated models. In Chapter 5, we will follow a learning-based approach while relying on data without annotations. This gives us the liberty to use any data available.

Feed-forward neural networks can approximate arbitrary functions $f^*(\mathbf{x}) = \mathbf{y}$ by learning a mapping $f(\mathbf{x}, \theta) = \mathbf{y}$ with θ the parameters of the network [GBC16]. A neural network usually consists of several layers each mapping output (activation) of the previous layer or the input of the network with a different function. For example, Fig. 2.2 shows a 2-layer neural network with densely connected layers. In a densely connected layer, each neuron passes the weighted sum over all neurons from the previous layer and a bias $\mathbf{b}_1, \mathbf{b}_2$ through a transfer function f_1, f_2:

$$\mathbf{h} = f_1(\mathbf{w}_1^T \mathbf{x} + \mathbf{b}_1) \tag{2.9}$$

$$\mathbf{y} = f_2(\mathbf{w}_2^T \mathbf{h} + \mathbf{b}_2) \tag{2.10}$$

In this case, the learned parameters are the weights and bias $\theta = (\mathbf{w}_1, \mathbf{w}_2, \mathbf{b}_1, \mathbf{b}_2)$. The simplest transfer (activation) function is the linear function $f(\mathbf{x}) = \mathbf{x}$. However, with this transfer function, only linear functions f^* can be represented. Therefore, non-linear functions are usually used instead in the hidden layers. The most popular activation function is the Rectified Linear Unit (ReLU), which allows for faster training than other non-linear functions such as the sigmoid:

$$\text{ReLU}(\mathbf{x}) = max(0, \mathbf{x}) \tag{2.11}$$

2.3.1 Learning

Training is performed using gradient descent with respect to the network weights θ. Usually, large datasets are used during training, and computing the gradients for the complete datasets is too expensive. Therefore, the gradients are approximated using stochastic gradient descent by iteratively sampling a random subset from the dataset (batch) of size N and computing the gradients with respect to this subset:

$$\theta = \theta + \mu \cdot \frac{1}{N} \sum_{i=1}^{N} \nabla \mathcal{L}(\hat{\mathbf{y}}_i, \mathbf{y}_i, \theta), \tag{2.12}$$

with $\mathcal{L}$ the loss function, $\hat{\mathbf{y}}_i$ the label, $\mathbf{y}_i$ the network output, and μ the learning rate.

Back-propagation efficiently computes the gradients of each layer using the chain rule. This way gradients and activations from proceeding layers can be reused in the computation of the gradients of the current layer:

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{h}_2} \frac{\partial \mathbf{h}_2}{\partial \mathbf{h}_1} \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}} \tag{2.13}$$

with $\mathbf{h}_1, \mathbf{h}_2$ the output of different hidden layers.

The learning rate affects the speed and convergence of the training. A too-large learning rate might lead to divergence, while a too-small learning rate might take too long to converge. Therefore, several approaches propose adapting the learning rate during training. AdaGrad [DHS11] adapts the learning rate of each parameter individually based on the sparsity of parameters. This way, extreme updates are reduced while small and less frequent updates are amplified. RMSProp [HSS12], in contrast, uses a running average over the gradients to adapt the learning rate. ADAM [KB15] extends this idea by taking the running average of the gradients and second moments of the gradients to adapt the learning rate.

2.3.2 Convolutional Neural Networks

Convolutional neural networks have been proposed to maintain spatial information when processing images [GBC16]. In contrast to standard neural networks, CNNs consider two-dimensional inputs $\mathbf{X}$, feature maps $\mathbf{H}$, and outputs $\mathbf{Y}$. The output of hidden layers in a CNN are usually referred to as feature maps. The most common layers are convolutional layers, which learn a kernel $\mathbf{K}$ of dimension (M, N) to convolve the input $\mathbf{X}$ at pixel (i, j):

$$\mathbf{H}(i,j) = (\mathbf{K} * \mathbf{X})(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \mathbf{X}(i-m, j-n) \cdot \mathbf{K}(m,n) \tag{2.14}$$

The kernel size is preset as well as the step size (stride) in which the convolution is applied. Convolutional layers have the useful property of equivariance, meaning that the output changes in the same way as the input. For instance, if we apply a shift operation onto the image and pass it through the convolution, it would be equal to passing the image through the convolution and applying the shift operator, afterwards. In optical flow, this is a desirable property since if we shift the input images, the optical flow should be shifted by the same amount but should not change beyond that.

Other common layers of CNNs are pooling, unpooling layers, and transposed convolutions. Pooling layers summarize neighborhoods of the input using some statistics, which results in a reduction of the input resolution. The most popular pooling layers are max and average pooling, which take the maximum or average from the neighborhood. Unpooling layers use stored indices from a previous pooling layer to approximately invert the pooling operation and restore the input resolution of the pooling layer. This is often used in encoder-decoder architectures [YHD16; Ren+17; PHC16; Wan+18b; MHR18] consisting of a contracting network reducing the resolution in each layer (encoder) and expanding network increasing the resolution (decoder). Transposed convolutions, also referred to as deconvolutions or fractionally strided convolutions, instead increase the input resolution by inserting zero activations between neurons of the input before convolving it with a kernel.

2.3.3 Regularization

One way to improve generalization of neural network is to use regularization. The different layers already enforce some kind of regularization on the network by applying specific operations. Another form of regularization is enforced on the parameters of the network. Weight decay jointly minimizes the loss function with the squared L2 norm of the network parameters. This favors solutions that use fewer features and should lead to focusing on important features. We will use weight decay in Chapter 5 to improve the generalization of our model.

Another popular and powerful way to regularize neural networks is by sharing weights of different branches in the network. Siamese network consists of two branches sharing the parameters to extract meaningful features from different inputs. This can be extended to an arbitrary amount of inputs, which will learn the same feature embedding for all inputs. We will be using this idea to extract features from multiple images in our network architecture (Section 5.2). In specific, we would like to learn the same feature embedding for all input images. This allows us eventually to match the features extracted of the different images to estimate the optical flow. In addition, to the strong regularization of the network, it also leads to a reduction of the parameters and, thus, faster training. While such networks can be trained independently with a triplet loss comparing a baseline input to a positive and negative example using a metric, we will be training our complete model end-to-end.

3 Related Work on Optical Flow

Considering two images, the reference $\mathbf{I}_0$ and target $\mathbf{I}_1$ image taken at time step $t = 0$ and $t = 1$, we are interested in the dense motion field $\mathcal{U}$ mapping each pixel $\mathbf{p}$ from $\mathbf{I}_0$ to a pixel in $\mathbf{I}_1$, as illustrated in Fig. 1.1. The estimation of optical flow was firstly addressed with a variational approach proposed by Horn and Schunck [HS81]. In concurrent work, Lucas and Kanade [LK81] proposed a local approach for the registration of image patches using a least-square formulation. Besides these differential approaches, frequency-based [Hee88] and phase-based approaches [FJ90] were also investigated that uses responses of different spatio-temporal filters to estimate the motion, but these approaches could not prevail against the differential approaches. Differential approaches were extended to handle large displacements [Ana89] and deal with outliers [BA93]. In addition, more general assumptions [SPC09; ZPB07; Mül+11; Pan12] and better optimization methods [Bro+04; SRB14] were presented. The introduction of sparse feature matching [BM11] allowed further improvements on large displacements and lead to novel approaches based on interpolation schemes [Rev+15], discrete optimizations [MHG15; GG16; CK16], and learning-based approaches [WB15]. Eventually, deep neural networks [Dos+15; RB16; Ilg+17; Sun+18b] were proposed to learn even better models directly from data. In this chapter, we will introduce the concepts of the different approaches in detail and relate them to the approaches presented in this thesis.

3.1 Variational Approaches

Variational methods rely on the calculus of variations discussed in Section 2.1 to estimate a continuous flow field $\mathcal{U}$. Towards this goal, they minimize global energy $\mathcal{E}(\mathcal{U})$ consisting of a data term $\mathcal{E}^{\mathcal{D}}$, measuring the photo-consistency, and a smoothness term $\mathcal{E}^{\mathcal{S}}$, encouraging similarity between spatial neighbors:

$$\mathcal{E}(\mathcal{U}) = \int_{\mho} \mathcal{E}^{\mathcal{D}}(\mathcal{U}, \mathbf{p}) + \mathcal{E}^{\mathcal{S}}(\mathcal{U}, \mathbf{p})\, d\mathbf{p} \tag{3.1}$$

with pixel $\mathbf{p} \in \mho$.

Horn and Schunck [HS81] assume that the intensity of a pixel is constant over time and propose the brightness constancy assumption:

$$\mathcal{E}^{\mathcal{D}}(\mathcal{U}, \mathbf{p}) = \rho\left(\mathbf{I}_0(\mathbf{p}) - \mathbf{I}_1(\mathbf{p} + \mathcal{U}(\mathbf{p}))\right) \tag{3.2}$$

with a penalty function ρ. A first-order Taylor expansion leads to a linearized version that holds for small displacements.

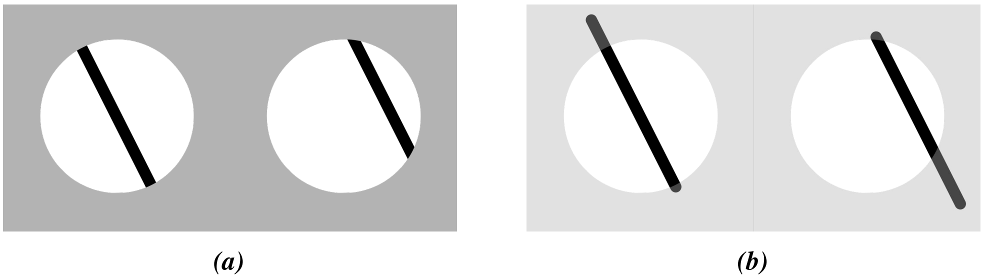


Figure 3.1: ***Aperture Problem*** *The motion of a moving line observed through a small aperture is ambiguous (a) since the motion component parallel to line cannot be observed (b).*

For a single pixel in isolation, this assumption yields one equation with two unknowns, which does not result in a unique solution. This problem is also known as the aperture problem in the flow literature and can be well illustrated with the following example. Let us consider a moving line visible through a small aperture, as illustrated in Fig. 3.1a. In this case, only the motion component perpendicular to the line can be observed, while the motion component parallel to the line could be arbitrary (Fig. 3.1b). In order to solve the aperture problem, additional constraints need to be introduced. The smoothness assumption is the most common constraint, which encourages similarity of spatially neighboring flow vectors as in Eq. (3.3). This assumption is motivated by the fact that objects in the real world usually follow a rigid motion or deformation. Therefore, neighboring pixels of the same object should have similar motion, and discontinuities typically occur only at object boundaries.

$$\mathcal{E}^{\mathcal{S}}(\mathcal{U},\mathbf{p}) = \rho\left(\frac{\partial \mathcal{U}}{\partial x}\right) + \rho\left(\frac{\partial \mathcal{U}}{\partial y}\right) \tag{3.3}$$

Variational optical flow formulations achieve high precision due to continuous optimization, especially in the case of small sub-pixel motions. Therefore, we follow a variational formulation for the estimation of the motion field between frames captured with a high frame rate in Section 4.2. We extend a variational formulation to multiple frames and jointly reason about occlusions. In Chapter 5, we use the brightness constancy assumption as supervision signal for learning optical flow estimation without ground truth. Joint learning of occlusions allows us to handle occluded regions, which violate the brightness constancy assumption.

3.1.1 Robustness

The original formulation [HS81] uses a quadratic penalty function in the data and smoothness term. Minimizing the squared error corresponds to maximizing the likelihood estimate while assuming a normally distributed error. This has strong limitations as violations of the brightness constancy (e.g., illumination changes) and smoothness assumption (e.g., discontinuities) cannot be handled. Black and Anandan [BA93] propose to replace the penalty function by a robust function such as the truncated quadratic or Lorentzian function. Robust penalty functions alleviate this problem by reducing the impact of outliers to zero but are also more difficult to optimize because of the non-convexity. In contrast, the Charbonnier penalty function turns out to be better suited for optimization while being more robust than the quadratic function [SRB14].

$$\rho\left(x\right) = \sqrt{x^2 + \varepsilon^2} \tag{3.4}$$

We will use the Charbonnier penalty in Chapter 4 and Chapter 5 to reduce the influence of outliers in the optimization.

3.1.2 Data Terms

While the robust function already reduces the impact of outliers, it does not resolve the problem of model violations. Illumination changes usually occur in real scenes and, therefore, new pixel- and patch-based data terms have been investigated that can better handle these violations. A popular pixel-based data term that can better handle illumination changes is the gradient constancy assumption. Instead of assuming that the brightness is constant over time, we assume that the image gradients in x (∇_x) and y (∇_y) direction are constant over time. In case of illumination changes, this assumption is more likely to hold since the image gradients will not be affected.

$$\mathcal{E}^{\mathcal{D}}\left(\mathcal{U},\mathbf{p}\right) = \rho\left(\nabla_x \mathbf{I}_0\left(\mathbf{p}\right) - \nabla_x \mathbf{I}_1\left(\mathbf{p} + \mathcal{U}\left(\mathbf{p}\right)\right)\right) + \rho\left(\nabla_y \mathbf{I}_0\left(\mathbf{p}\right) - \nabla_y \mathbf{I}_1\left(\mathbf{p} + \mathcal{U}\left(\mathbf{p}\right)\right)\right) \tag{3.5}$$

Patch-based data terms such as the normalized cross correlation [SPC09], mutual information [Pan12] and census transform [Mül+11] compare image statistics of small patches centered around a pixel. However, the optimization for a joint occlusion reasoning becomes cumbersome since the occlusion states of each pixel need to be taken into account while computing the statistics. A simplification would model the occlusion of a patch by the occlusion of the center pixel. In our experiments, the joint occlusion reasoning worked better with the gradient constancy assumption. Therefore, we use this assumption in Section 4.2.1, Section 4.3.1, and Section 5.3.1 when dealing with real data in combination with the brightness constancy assumption.

3.1.3 Regularization

Flow discontinuities frequently occur near motion boundaries caused by objects moving in front of each other. The original formulation by Horn and Schunck [HS81], cannot handle these discontinuities due to a homogeneous, non-robust smoothness term. Total Variation (TV) regularization used in Zach, Pock, and Bischof [ZPB07] replaces the quadratic penalization by the L_1 norm to preserve discontinuities in the flow field. In addition, the image gradients are often used to reduce the smoothness term in case of image gradients since they often occur with object boundaries. We use such regularization terms in Chapter 4 and Chapter 5 to encourage similar motion between neighbors and propagate information into ambiguous regions.

However, like the original formulation by Horn and Schunck, this model also biases the solution towards fronto-parallel surfaces leading to artifacts in the estimation results, in particular in the presence of strongly slanted planes (e.g., the road surface). Thus, higher-order regularizations like the Total Generalized Variation (TGV) model have been proposed [BKP10]. TGV priors can better represent real data as they leverage a piecewise affine motion model. The non-local Total Generalized Variation [RBP14] is an extension of this model, which enforces the piecewise affine assumption in a local neighborhood. This allows them to improve the performance in regions where the data term is ambiguous in comparison to TGV, which considers only direct neighbors. Especially in street scenes that we consider in Chapter 5, the TV regularization is often violated. Therefore, we replace the first-order regularization when training on street scenes by a second-order regularization. We enforce the second-order regularization only on neighboring pixels since we use simple gradient descent for training and more sophisticated optimization techniques are necessary for larger neighborhoods.

3.2 Large Displacements

One major challenge, in particular for variational methods, is the estimation of large displacements since linear approximations are used that only hold in the case of pixel motion. In variational formulations, this problem is typically addressed with a coarse-to-fine strategy [Ana89; BA96; Bro+04], estimating the flow on a coarser resolution to initialize the estimation on a finer resolution. These iterative optimization schemes use a warping function to transform the target image according to the current optical flow estimate with bilinear interpolation to handle sub-pixel precision. Thus, only the residual flow field between $\mathbf{I}_0$ and $\mathbf{I}'_1$ needs to be estimated in each iteration with

$$\mathbf{I}'_1(\mathbf{p}) = \mathbf{I}_1(\mathbf{p} + \mathcal{U}(\mathbf{p})). \tag{3.6}$$

While this strategy works for large structures of little complexity by capturing the dominant motion in the scene, fine geometric details are often lost in the process. Besides, textural details important for correspondence estimation are lost at coarse resolutions, hence leading the optimizer to a local minimum. These problems can be alleviated by integrating sparse feature correspondences into the variational formulation as proposed by Brox and Malik [BM11]. The feature matches, obtained from nearest neighbor search on a coarse grid, are used as soft constraint in a coarse-to-fine optimization. Revaud et al. [Rev+15] go one step further and completely replace the coarse-to-fine strategy by an interpolation of sparse matches as initialization of the dense optimization at full resolution. They propose to use the geodesic distance for interpolation, which is aware of image edges. Sparse matches are obtained using DeepMatching, a deep neural network matching approach introduced by Weinzaepfel et al. [Wei+13].

In addition, discrete optimization methods [MHG15; GG16; CK16] were proposed to address large displacements. Menze, Heipke, and Geiger [MHG15] use Approximate Nearest Neighbor (ANN) search to generate a set of proposals as candidates to be used in a discrete optimization framework. Inference is made feasible by restricting the number of matches to the most likely ones and by exploiting the truncated form of the pairwise potentials. Motivated by the success of Siamese networks in stereo [ŽL16], Güney and Geiger [GG16] extend this work to learning features for 2D patch matching. They further investigate the importance of the receptive field size exploiting dilated convolutions as proposed by Yu and Koltun [YK16] for semantic segmentation. While several works [MHG15; GG16] use pruning to make inference feasible, Chen and Koltun [CK16] propose a discrete optimization over the full space. Min-convolutions [FH12; Che+14] are used to reduce the complexity and to effectively optimize the large label space using a modified version of TRW-S [Kol06].

A more sophisticated search strategy than ANN was proposed by Bailer, Taetz, and Stricker [BTS15; BTS17]. They propose a hierarchical search scheme that addresses the correspondence problem on different scales, similar to the coarse-to-fine scheme of variational methods. While simple ANN has many outliers due to missing regularization, they rely on spatial propagation and random search to reduce the number outliers. The remaining outliers are removed based on a consistency check between flow fields, and the interpolation scheme presented by Revaud et al. [Rev+15] is used to fill the resulting gaps. Schuster et al. [Sch+18] extend the approach with a novel interpolation scheme that detects edges more robust than Revaud et al. [Rev+15] using a random forest.

Wulff and Black [WB15] present a different approach to obtain dense optical flow from sparse matches. In their approach, the optical flow field is represented as a weighted sum of basis flow fields learned from reference flow fields, which have been estimated from Hollywood movies. They estimate the optical flow by finding the weights that minimize the error with respect to the detected sparse feature correspondences. While this results in overly smooth flow fields, the so called PCA Flow approach is very fast compared to variational and discrete optimization methods. A slower but more accurate version based on a layered representation of the scene is also proposed to better handle flow discontinuities.

In Section 4.2.1, we consider HFR sequences that mostly follow small motions. Therefore, we rely on an initialization obtained from Revaud et al. [Rev+15] followed by a variational coarse-to-fine method. This allows us to strongly reduce the number of scales used during the optimization while obtaining good estimates for larger motions. In contrast, the dense tracking formulation discussed in Section 4.3.1 relies on a discrete optimization over a set of proposal trajectories to make the optimization feasible.

3.3 Classic Multi-Frame Optical Flow

While the majority of optical flow methods use two input frames, few works have exploited the properties of temporal coherence in video sequences. The early frequency and phase-based approaches [Hee88; FJ90; GH02] apply spatio-temporal filters on space-time volumes spanned by a video sequence. Edges in the temporal domain correspond to the motion vector of the corresponding brightness patches and can be detected using a predefined set of filters. Similarly, [BB87] propose epipolar-plane image analysis to recover the rigid camera motion from imagery that is dense in time.

Simple variational formulations incorporate temporal information [WS01; SVB13; ZBW11; RDR13] with a penalty on the magnitude of flow gradients. These methods only work for very small motions and a small number of frames as the change of location is not taken into account. Several works [Vol+11; SS07; SSB10] incorporate constant velocity priors into the variational optical flow estimation process. A constant acceleration model has been used by Black and Anandan [BA91] and Kennedy and Taylor [KT14] and layered approaches have been proposed by Sun et al. [Sun+13] and Sun, Sudderth, and Black [SSB12]. Wang, Fan, and Wang [WFW08] proposed a general multi-frame extension for local optimization methods. They suggest two different data terms one comparing the reference image to all others and the other comparing all successive frames. In their formulation any motion model can be used but they use a constant velocity model in the experiments. We incorporate both data terms in our variational formulation (Section 4.2.1) for the estimation of the optical flow in HFR sequences. We also rely on a constant velocity model but, in contrast, we jointly estimate occlusions and use them to weight our data terms accordingly.

Unfortunately, none of the methods mentioned above are directly applicable to our data generation problem discussed in Chapter 4, which requires dense pixel tracking through large space-time volumes. Lim and Gamal [LG01] and Lim, Apostolopoulos, and Gamal [LAG04; LAG05] use Lucas-Kanade algorithm on the high frame rate and combine the estimation along the trajectories to obtain the low frame rate flow. While they show the benefit of temporally oversampling for optical flow estimation, they also observe that motion aliasing has a strong impact on the accuracy of the flow estimation and arises not only from high frequency motions but also from high spatial frequencies. Sand and Teller [ST08] combine sparse optical flow between frames with long range tracking. However, the approach is computationally expensive and can, therefore, not be applied densely. In contrast, we follow a two-stage approach: We first estimate temporally local flow fields and occlusion maps using a novel discrete-continuous multi-frame optimization, exploiting linearity within small temporal windows. We expect that most objects move approximately with constant velocity over short time intervals due to the physical effects of mass and inertia. Second, we reason about the whole space-time volume based on these predictions.

Multi-frame formulations are also better suited for reasoning about the visibility of pixels. In a simple two frame formulation, image information is only provided for visible pixels and occlusions can only be detected by large inconsistency of the image regions. By considering additional frames (past and future), information about the actual motion of the pixel can be recovered and used for better occlusion reasoning. Therefore, we use this property in Chapter 4 to improve the optical flow estimation in occluded regions and obtain sharper motion boundaries. Occluded regions are even more problematic in unsupervised learning of optical flow due to the weak photometric terms used for training. Therefore, we leverage a multi-frame formulation in Chapter 5 to learn optical flow and occlusions in an unsupervised fashion. More specifically, we focus on the minimal case of three frames, which allows us to reason about the visibility of a pixel while expecting only little appearance changes that mostly adhere to the brightness constancy assumption.

3.4 Deep Learning for Optical Flow

Most optical flow approaches do not incorporate high-level information making it hard to overcome ambiguities that require reasoning about larger image regions. A few notable exceptions are methods incorporating semantic information into their formulation [Bai+16; Sev+16] and layered approaches [Sun+13; SSB12; YMU13; YMU14]. Convolutional neural networks are able to learn high-level assumptions from data directly. In contrast to previous formulations, more sophisticated models can be learned that better represent the real world.

However, the limited amount of annotated data hindered the development of deep learning approaches until Dosovitskiy et al. [Dos+15] presented the large-scale synthetic dataset Flying Chairs. They created the dataset by rendering 3D chair models on top of images from Flickr. With the dataset, they proposed FlowNet to learn optical flow end-to-end using a CNN. FlowNet consists of a contracting part that extracts important features and an expanding part that produces the high resolution optical flow field as output. They propose two different architectures: a simple network (FlowNetSimple) stacking the images and a complex network (FlowNetCorr) correlating features of the separately processed images. This first attempt to learning optical flow end-to-end demonstrated that it was possible to learn optical flow estimation from data, despite not yet reaching the performance of state-of-the-art traditional methods on KITTI or Sintel. However, due to the parallel Graphical Processing Unit (GPU) implementation, FlowNet was able to run in real time as opposed to most of the classical algorithms implemented on the CPU.

In contrast to the contracting and expanding networks, Ranjan and Black [RB16] present SPyNet, an architecture inspired by the coarse-to-fine strategy leveraged in traditional optical flow estimation techniques. Each layer of the network represents a different scale and only estimates the residual flow with respect to the image warped according to the flow of the previous layer. This formulation allowed them to achieve similar performance as FlowNet while being faster and 96 % smaller in terms of network weights, making it attractive for embedded systems with limited compute capabilities. Ilg et al. [Ilg+17] present FlowNet2, an improved version of FlowNet, by stacking the architectures and fusing the stacked network with a subnetwork specialized on small motions. Similar to SPyNet, they also input the warped image into the stacked networks. Each stacked network estimates the flow between the original frames instead of the residual flow as in SPyNet. In contrast to FlowNet and SPyNet, they use the FlyingThings3D dataset [May+16] consisting of 22k renderings of static 3D scenes with moving 3D models from the ShapeNet dataset [SCH15]. PWC-Net [Sun+18b] combines the classical ideas of coarse-to-fine warping [RB16] and cost volume filtering [Dos+15] with a Siamese network that proved to learn rich feature representations [ŽL16]. This combination of classical ideas into the network architecture allows them to achieve state-of-the-art performance with a small number of network weights. Recently, Hur and Roth [HR19] suggested an iterative residual refinement scheme for FlowNet2 and PWC-Net inspired by classical optimization methods. They propose to apply backbone networks in an iterative fashion while sharing the weights of the networks. In each iteration, a residual estimation problem will be addressed by warping the target image according to the previous optical flow.

3.4.1 Unsupervised Learning

The dependency of deep neural networks on large annotated datasets has recently motivated the development of unsupervised learning techniques. Impressive results have been demonstrated for single image depth prediction [Gar+16; XGF16; GMB17; Zho+17; Vij+17], ego-motion estimation [ACM15; Zho+17; Vij+17] and optical flow [PHC16; YHD16; Lon+16; All+17; Vij+17; Ren+17; Wan+18b; MHR18].

In a typical unsupervised optical flow framework, a photometric loss is used in combination with a smoothness loss for untextured regions [PHC16; YHD16; Lon+16; All+17; Vij+17; Ren+17; Wan+18b; MHR18; Ran+19a]. More specifically, the target image is warped according to the predicted flow and compared to the reference image using a photometric loss. Typically, an encoder-decoder network [YHD16; Ren+17; PHC16; Wan+18b; MHR18] is used. Pătrăucean, Handa, and Cipolla [PHC16] combine the simple encoder-decoder network with a convolutional LSTM to incorporate information from previous frames. For unsupervised learning of optical flow, single view depth, camera motion, and semantic segmentation, Ranjan et al. [Ran+19a] present a new framework called competitive collaboration. In the spirit of expectation-maximization, a set of neural networks act as competitors for describing the motion of the static and dynamic part of the scene while a moderator network assigns each pixel to be either static or dynamic. In an iterative procedure, first, the competitors are trained based on the current assignment by the moderator. Then, the moderator is trained based on the current ability of the competitors to explain the different types of motion.

Recently, several approaches [MHR18; Wan+18b] noticed that occluded regions introduce errors in the photometric loss that cause misleading gradients during training. They propose to mask out occluded regions in order to avoid this problem. While both of them jointly learn the forward and backward flow, Meister, Hur, and Roth [MHR18] use a forward-backward consistency check and Wang et al. [Wan+18b] create a range map with the backward flow, counting the correspondences for each pixel in the reference frame. However, both approaches use a heuristic to obtain the final occlusion map. Another recent work on unsupervised learning of depth and ego-motion [Zho+17] predicts explainability masks to exclude dynamic objects and occlusions using a photometric loss function. While [Zho+17] only addresses static scenes, we target the general unconstrained optical flow problem and learn to jointly predict flow and occluded regions in this setting.

In contrast to the heuristics used in [MHR18; Wan+18b], we propose to jointly learn the optical flow and occlusions in Chapter 5. We relate flow and occlusion estimates in our photometric loss by weighting information from the future and the past according to occlusion estimates. This joint formulation allows us to train our occlusion-aware model from scratch in contrast to Meister, Hur, and Roth [MHR18] that requires pre-training without occlusion reasoning.

3.5 High Speed Flow

With some exceptions (Wulff and Black [WB15], Timofte and Gool [TG15], Weinzaepfel et al. [Wei+13], Farneback [Far03], and Zach, Pock, and Bischof [ZPB07]), most of the

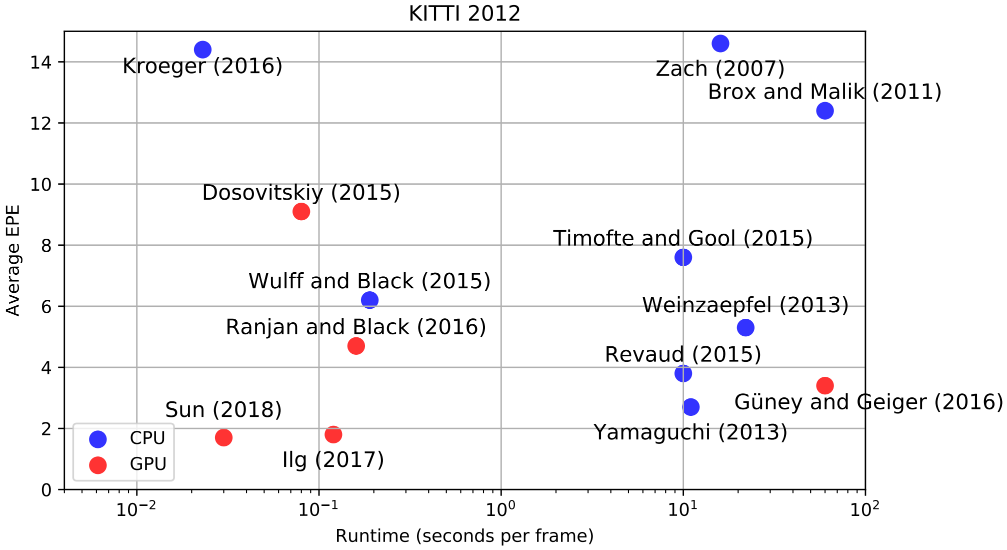


Figure 3.2: Accuracy vs Efficiency. *Trade-off between performance and speed on KITTI 2012 [GLU12].*

classical optical flow approaches are very inefficient and cannot be applied in real-time, which is necessary for applications such as autonomous driving. The trade-off between accuracy and speed for different algorithms on the KITTI 2012 benchmark [GLU12] is illustrated in Fig. 3.2.

While variational approaches yield a good precision, they belong to the slowest set of methods for motion estimation. The duality-based approach for total variation optical flow proposed by Zach, Pock, and Bischof [ZPB07] allows an efficient GPU implementation that performs in real-time (30 Hz) on a resolution of 320×240. Sparse matching approaches are usually more efficient than variational formulations but often need variational refinement as post processing step to achieve sub-pixel precision. The approach proposed by Kroeger et al. [Kro+16] allows to trade-off accuracy and run-time. They obtain fast patch correspondences with inverse search resulting in a dense flow field when aggregating patches across multiple scales. This allows them to estimate optical flow at up to 600 Hz but at the cost of accuracy. The recent introduction of deep learning to the optical flow problem yielded several near real-time approaches (Dosovitskiy et al. [Dos+15] and Ranjan and Black [RB16]) including (Ilg et al. [Ilg+17] and Sun et al. [Sun+18b]), which achieve state-of-the-art performance on popular datasets.

In the generation of reference data discussed in Chapter 4 the efficiency is not crucial since the method is applied offline. However, we rely on sparse matches to improve the efficiency when estimating the motion between HFR frames. In contrast, our unsupervised learning scheme (Chapter 5) is applied to a network based on PWC-Net that achieves near real-time performance.

3.6 Confidence Measures

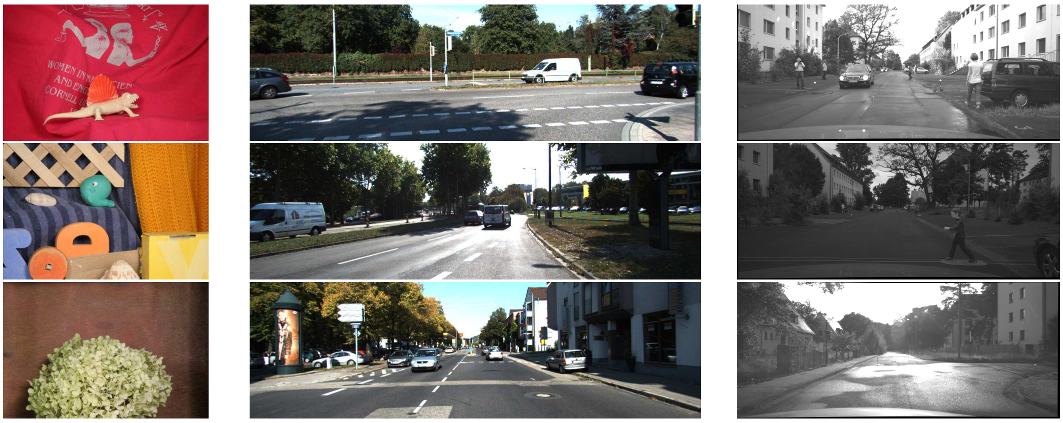
A confidence measure to assess the quality of the estimated flow is desirable, considering the remaining challenges in optical flow. For instance, in applications that use optical flow estimates such as action recognition, the importance of these estimates can be adjusted with a good confidence measure. Bad flow estimates would have a lower impact, while good estimates would have a high impact.

Several measures based on spatial and temporal gradients have been proposed [Ura+88; Ana89; SAH91] to quantify the uncertainty in the optical flow estimate. In contrast, algorithm-specific measures propose confidence estimates for a specific group of methods, i.e., variational methods [BW06] and general methods for pixel-based minimization problems [KN11]. While Bruhn and Weickert [BW06] propose a confidence measure based on the energy function optimized by the variational method, Kybic and Nieuwenhuis [KN11] uses bootstrap resampling, which repeatedly run the optical flow computation while randomly replacing the contributions of some pixels to the energy.

Learning-based measures [Kon+07; KMG08; Mac+13] learn a model that relates the success of flow algorithm success to spatio-temporal image data or the computed flow field. Kondermann et al. [Kon+07] use linear subspace projection of the optical flow and define a confidence based on the reconstruction error using the linear basis. In contrast, Kondermann et al. [Kon+07] learn a probabilistic motion model from annotated training data and use hypothesis testing of flow estimates based on the derived model to compute confidences. Mac Aodha et al. [Mac+13] learn a classifier to directly measure the quality of the optical flow predictions based on multiple feature types, such as temporal features, texture or distance from image edges.

Several approaches [WKR17; Ilg+18; GR18] proposed to estimate the optical flow and confidences simultaneously. Wannenwetsch, Keuper, and Roth [WKR17] formulate a probabilistic method based on general energy formulations. The optical flow is estimated by minimizing the expected loss over the posterior, while confidences are measured using the marginal entropy of the posterior. They rely on a mean-field approximation to make inference tractable. Recently, Gast and Roth [GR18] propose lightweight probabilistic CNNs. Instead of learning a two-dimensional optical flow field, they learn the mean and standard deviation of a Gaussian distribution. Furthermore, they suggest to learn distribution in each layer and describe how to propagate the probabilistic activations in forward and backward direction. In concurrent work, Ilg et al. [Ilg+18] suggest two approaches for learn uncertainties using CNNs. In a simple approach, they train a set of different models and estimate uncertainty empirically. Since training several models is expensive, they propose an extension of FlowNet [Dos+15] in the spirit of [GR18] by replacing some optical flow layers by the mean and standard deviation of a Gaussian.

In the generation of new reference data discussed in Chapter 4, such a measure would allow us to weigh the importance of each estimate according to the confidence. This is beneficial when using the reference data for evaluation or training of methods. In this thesis, we focus our attention on the challenging estimation problem and will only rely on heuristics to remove regions where our approach fails. However, a probabilistic extension that jointly estimates the optical flow and confidence similar to the works [Ilg+18; GR18] would be of



(a) Middlebury *(b) KITTI 2015* *(c) HCI Benchmark*

Figure 3.3: **Real Datasets.** *Examples from the real optical flow datasets Middlebury [Bak+11], KITTI 2015 dataset [Gei+13; MG15], and MCI Benchmark [Kon+16].*

great interest for future studies.

3.7 Datasets

The acquisition of optical flow ground truth is very difficult since no sensor exists that can capture optical flow ground-truth in natural scenes. Thus, there are only a few real datasets for the optical flow problem. In Fig. 3.3, we show three examples for each dataset.

The first unified optical flow benchmark Middlebury was proposed by Baker et al. [Bak+11] providing a test environment and evaluation server for optical flow approaches. The benchmark consists of sequences with non-rigid motion, synthetic sequences and a subset of the Middlebury stereo benchmark sequences (static scenes). For all non-rigid sequences, ground truth flow is obtained by tracking hidden fluorescent textures sprayed onto the objects. The process is very time consuming and cannot be applied on scenes outside of the laboratory. Therefore, the Middlebury dataset is limited in size and missing real world challenges like complex structures, lightning variation and shadows. In addition, Middlebury only contains small motions of up to twelve pixels, which do not allow the investigation of challenges related to fast motions.

In contrast, the KITTI Benchmark introduced by Geiger, Lenz, and Urtasun [GLU12] and Geiger et al. [Gei+13] provides optical flow ground truth for real street scenes. The dataset has been captured from an autonomous driving platform equipped with senor suite consisting of high-resolution cameras and a Velodyne 3D laser scanner. They obtain sparse ground truth for the static part of the scene by projecting accumulated 3D laser point clouds onto the images. In KITTI 2015 [MG15], the ground truth for vehicles is added by fitting accurate 3D CAD models to all vehicles in motion. Both KITTI 2012 and 2015 comprise 194 training and 195 test image pairs at a resolution of 1280 × 376 pixels each. However, a multi-view extension of the dataset is provided consisting of approximately 4000 images without annotations. While KITTI provides annotated data and an evaluation server, it is still limited in size for deep learning. Therefore, the KITTI dataset is usually used for evaluation and fine-tuning.

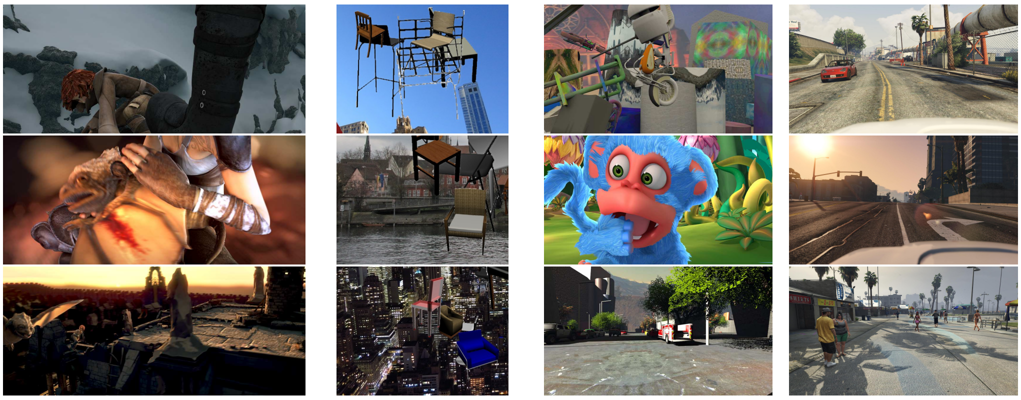
Similar to KITTI, Kondermann et al. [Kon+16] present the HCI Benchmark, an optical flow dataset and online benchmark for street scenes consisting of 28,504 image pairs. The benchmark specifically includes realistic, systematically varied radiometric and geometric challenges for autonomous driving. In contrast to the mobile laser scanning solution of KITTI, the static scene is scanned only once using a high-precision laser scanner in order to obtain a dense and highly accurate ground truth of all static parts. However, ground truth for dynamic objects is missing and dynamic regions are manually masked out. The major limitation of the HCI Benchmark is that all sequences were recorded in a single street section, thus lacking in diversity. While this enabled better control over the content and environmental conditions, it is the major reason why the datasat is still rarely used in the optical flow literature.

The proposed techniques for the generation of optical flow ground truth have several advantages and disadvantages. Middlebury's approach based on fluorescent textures provides very accurate ground truth but is restricted to a lab environment and needs a time consuming preparation. In contrast, KITTI generates ground truth outside of the lab. However, the re-projection of laser measurements into the images only allow for sparse ground truth and the setup is not applicable in arbitrary environments. In addition, cars are the only class of dynamic objects where approximate ground truth is provided. Finally, the technique used in HCI Benchmark can further improve on the precision of the optical flow ground truth in comparison to KITTI but is restricted to a certain area that was scanned in advance. In conclusion, all real datasets so far are restricted to a certain environment or setting and are missing complex scenes with non-rigid objects. We tackle this problem in Chapter 4 with a novel approach to obtain accurate reference data from High-Speed video cameras by tracking pixel through densely sampled space-time volume. In contrast to previous methods, our approach allows the acquisition of optical flow ground truth in challenging everyday scenes and, in addition, to augment the data with realistic effects such as motion blur to compare methods in varying conditions. Using this approach, we generate 160 diverse real-world sequences of dynamic scenes with a significantly larger resolution (1280 × 1024 pixels) than previous optical datasets and compare several state-of-the-art optical techniques on this data under varying conditions.

3.7.1 Synthetic Datasets

The problem of acquiring optical flow ground truth can also be resolved by creating synthetic datasets. While the synthetic optical flow datasets provide many examples for training deep neural networks, they lack in realism and are limited in diversity, as can be observed in Fig. 3.4. Therefore, large-scale synthetic datasets are usually used for pre-training and, afterwards, the pre-trained models are fine-tuned on small, more realistic datasets.

Butler et al. [But+12] take advantage of the open source movie Sintel, a short animated



(a) MPI Sintel *(b) Flying Chairs* *(c) Flying Things* *(d) Playing for Ben.*

Figure 3.4: **Synthetic Datasets.** *Examples from the synthetic optical flow datasets MPI Sintel [But+12], Flying Chairs [Dos+15], Flying Things [May+16], and Playing for Benchmark [RHK17].*

film. They create the MPI Sintel optical flow benchmark[1] by rendering scenes with optical flow ground truth. Sintel consists of 1,628 frames and provides three different datasets with varying complexity that are obtained using different passes of the rendering pipeline. Similar to Middlebury, they provide an evaluation server for comparison.

The limited size of optical flow datasets hampered the training of deep high-capacity models. Thus, Dosovitskiy et al. [Dos+15] introduced a simple synthetic 2D dataset of flying 3D chairs rendered on top of random background images from Flickr to train a CNN. As the limited realism of this dataset proved insufficient to learn highly accurate models, Mayer et al. [May+16] presented another large-scale dataset consisting of three synthetic stereo video datasets with optical flow ground truth: FlyingThings3D, Monkaa, Driving. FlyingThings3D provides everyday 3D objects flying along randomized 3D trajectories in a randomly created scene. Inspired by the KITTI dataset, a driving dataset has been created, which uses car models from the same pool as FlyingThings3D and additionally highly detailed tree and building models from 3D Warehouse. Monkaa is an animated short movie similar to Sintel used in the MPI Sintel benchmark.

Recently, powerful game engines have been used to generate synthetic datasets. In Playing for Data, Richter et al. [Ric+16] extract pixel-accurate semantic label maps for images from the commercial video game Grand Theft Auto V. This work was extended in [RHK17] to obtain dense correspondences from the game engine. Towards this goal, they developed a tool, which operates between the game and graphics hardware. Their algorithm allows them to produce dense optical flow annotations for around 250,000 images synthesized by the photo-realistic open-world computer game with minimal human supervision. They provide an evaluation server and split the dataset into a training, validation, and test set consisting of 134K, 50K, and 70K frames.

[1] `http://sintel.is.tue.mpg.de/`

4 Generating Reference Flow with High-Speed Cameras

The generation of pixel-level annotations is very laborious for most computer vision tasks. Only a few tasks such as 3D reconstruction can directly obtain ground truth from sensors (Kinect, LiDAR). Most tasks rely on manual annotations and allow the distribution of the work with crowdsourcing solutions such as Amazon Mechanical Turk [Ama]. However, no sensor exists that directly captures optical flow ground truth, and the dense manual annotation of subpixel accurate motion is unfeasible.

As a consequence, less training data is available, preventing progress in learning-based optical flow methods. While Middlebury [Bak+11] or KITTI [GLU12; MG15] provide real examples with ground truth, both datasets are very limited in size and diversity. Middlebury was recorded in a lab setting, and KITTI only consists of street scenes. Synthetic datasets [But+12; Dos+15; May+16] provide an attractive alternative to real images. However, the generation of synthetic datasets requires detailed 3D models and, thus, sometimes faces legal issues [Ric+16]. In addition, it remains an open question whether the realism and variety attained by rendered scenes are sufficient to match the performance of models trained on real data.

Special setups have been used to track pixels densely over time in image sequences. While Middlebury used fluorescent ink in combination with UV-light to obtain dense correspondences, KITTI used a LiDAR laser scanner to track the pixels of the static scene. However, both approaches are somewhat limited in the scenarios where they can be used. For a diverse and realistic dataset, a setup would be desirable that can be used in any condition. Therefore, we propose to exploit the power of high-speed video cameras for creating accurate optical flow reference data in a variety of natural scenes, see Fig. 4.1. In High-Frame-Rate (HFR) sequences the optical flow problem is much simpler because of smaller motion magnitudes (smaller search space) and minor appearance changes.

The recent advances in visual sensing hardware that is able to record high frame rates lead to many hand-held high-speed cameras. Current consumer cameras like the iPhone (since Model 6 [1]) or the GoPro (since model 4 [2]), for instance, are able to shoot 1 Megapixel videos at up to 240 fps. Besides the advancements in traditional camera technology, event-based vision sensors [Mue+15; Kim+14; MGS15] are emerging that transmit only sparse differential intensity information and, thus, have the potential of increasing frame rates even further up to the physical transmission limits. For our dataset, we use the Fastec TS5Q camera[3]. In contrast to consumer cameras, the Fastec is able to record QuadHD (2560 ×

[1] http://www.apple.com/de/iphone-6/specs

[2] https://shop.gopro.com/hero4/hero4-black/CHDHX-401.html

[3] http://www.fastecimaging.com/products/handheld-cameras/ts5

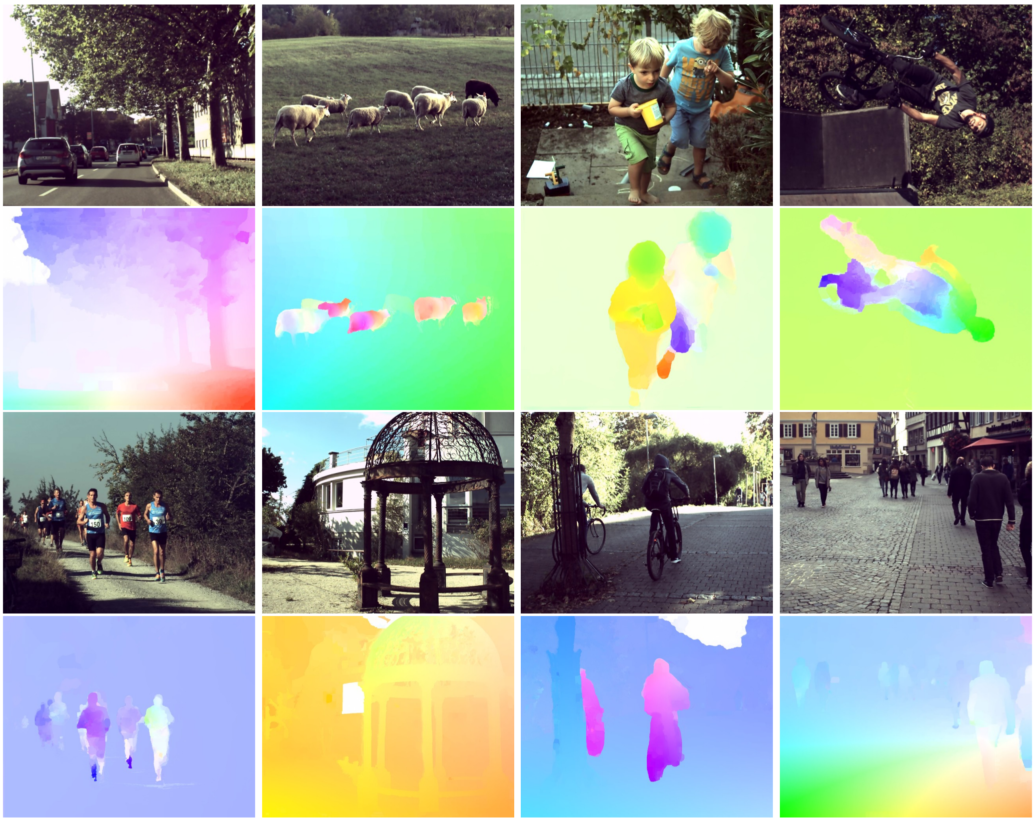


***Figure 4.1:* Illustration.** *This figure shows reference flow fields with large displacements established by our approach. Saturated regions (white) are excluded in our evaluation.*

1440 Pixels) videos with up to 360 fps. In addition, we do not need a special setup that might restrict the usage of the camera since it is a hand-held camera with external memory.

We record videos at high spatial and temporal (> 200 fps) resolutions and propose a novel approach to predict very accurate correspondences at regular spatial and temporal resolutions. Towards this goal, we track pixels densely over a large number of high-resolution input frames. The high spatial resolution provides fine textural details while high temporal resolution ensures small displacements allowing the integration of strong temporal constraints. Unlike Middlebury [Bak+11], our approach does not assume special lighting conditions or hidden texture. Compared to KITTI [GLU12; MG15], our method applies to non-rigid dynamic scenes, does not require a laser scanner, and provides dense estimates. In addition, our approach allows for realistically altering the input images, e.g., by synthesizing motion blur as illustrated in Fig. 4.14.

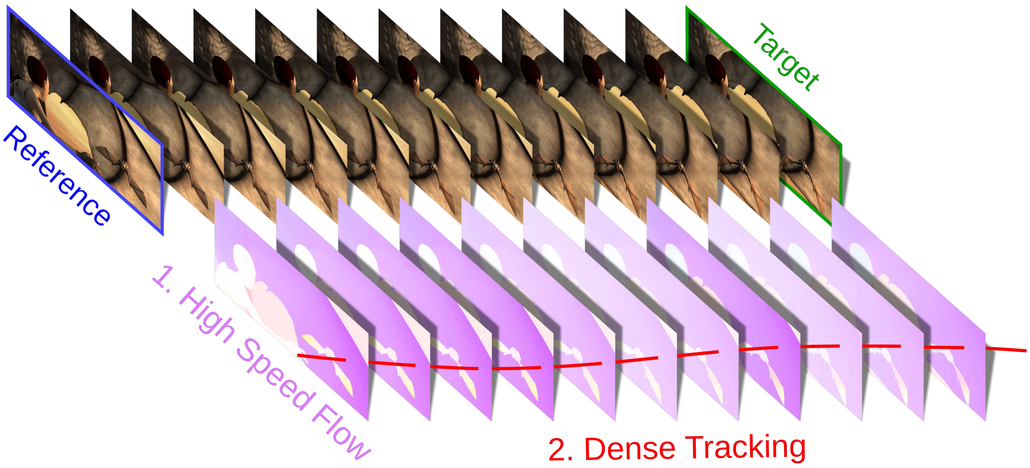


Figure 4.2: **Slow Flow Formulation.** *We address the hard problem of dense pixel tracking through the space-time volume by splitting the problem into many simpler problems, namely the motion estimation between intermediate frames (1. High-Speed Flow). Finally, we use the intermediate solutions to solve the tracking problem (2. Dense Tracking).*

4.1 Formulation

Let us consider a HFR video sequence $\mathcal{I} = \{\mathbf{I}_1, \ldots, \mathbf{I}_N\}$ consisting of N image frames $\mathbf{I}_t \in \mathbb{R}^{w \times h \times c}$ of resolution $w \times h$ and c channels. Besides the color intensities from the input sequence, we also consider the image gradients $\frac{\partial \mathbf{I}}{\partial x}$ and $\frac{\partial \mathbf{I}}{\partial y}$, as proposed by [Bro+04]. In contrast to the image intensities, the image gradients are less affected by illumination changes and therefore lead to more robust data terms. Similar to previous approaches, we control the influence of the image gradients with a weight ω_G we multiply to the channels. This results in $c = 9$ feature channels for each image $\mathbf{I}_t$ in total.

Our final goal is to estimate the optical flow $\mathbf{U}_{1 \to N}$ from frame 1 to N, exploiting all intermediate frames. Direct optimization of the full space-time volume is expensive and hard since it involves many unknown variables and a highly non-convex energy function. Therefore, we split the task into two simpler problems, as illustrated in Fig. 4.2:

1. “Flowlets”: We first estimate very accurate small-displacement flow fields $\{\mathcal{U}_{t \to t+1}\}$ between intermediate frames as described in Section 4.2

2. “Dense Tracking”: We formulate a dense tracking problem in Section 4.3, which uses the Flowlets to estimate the final flow field $\mathbf{U}_{1 \to N}$

4.2 Multi-Frame High-Speed Flow

First, we would like to discuss how to accurately estimate the optical flow between intermediate images of a HFR sequence. Given the HFR, the motion estimation problem between two intermediate frames is much simpler. On the one hand, we can expect that only

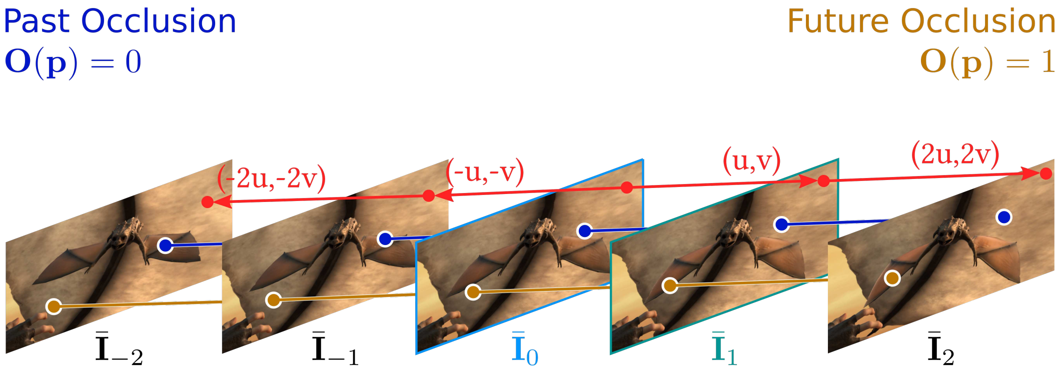


Figure 4.3: **Flowlets Formulation.** *Illustration of the linear hard constraint and occlusions variables for $T = 2$. While the blue pixel is occluded in the past, the brown pixel is occluded in the future. Note that by definition, all pixels are visible in the reference frame $\bar{\mathbf{I}}_0$.*

small motions will occur and thus only need to consider a reduced search space. On the other hand, only small changes occur in the scene because of the short time period between intermediate frames. Thus, simple assumptions such as the brightness constancy or linear motion assumptions are more likely to hold.

The Flowlets will be used as input to our dense tracking formulation. While the optical flow between intermediate frames is small, we would like to track pixels over longer periods, which will eventually result in large motions. Thus, small errors in the Flowlets could potentially accumulate over time to a large drift. As discussed in Section 3.1, variational approaches are among the most accurate approaches for optical flow estimation. We use a variational formulation to tackle the Flowlets estimation and alleviate the drift problem. However, if we use a classical variational approach to compute the Flowlets and naively combine them with a summation along the trajectory, we obtain large drift in visible regions and occlusions, as can be observed in Fig. 4.4. Therefore, we extend the formulation to multiple frames and jointly reason over occlusions.

Let $\{\bar{\mathbf{I}}_{-T}, \ldots, \bar{\mathbf{I}}_0, \ldots, \bar{\mathbf{I}}_T\}$ with $\bar{\mathbf{I}}_t = \mathbf{I}_{s+t}$ denote a short window of images from the video clip (e.g., $T = 2$), centered at reference image $\bar{\mathbf{I}}_0 = \mathbf{I}_s$. For each pixel $\mathbf{p} \in \mho \subset \mathbb{R}^2$ in the reference image $\bar{\mathbf{I}}_0$ we are interested in estimating a continuous function $\mathcal{U}(\mathbf{p}) = (u, v) \in \mathbb{R}^2$ that describes the displacement of $\mathbf{p}$ from frame $t = 0$ to $t = 1$ as well as an occlusion map $\mathbf{O}(\mathbf{p}) \in \{0, 1\}$ where $\mathbf{O}(\mathbf{p}) = 1$ indicates that pixel $\mathbf{p}$ is occluded in the future (i.e., occluded at $t > 0$).

Modeling all possible occlusions states for each pixel would add $h \cdot w \cdot 2 \cdot T$ unknown binary variables to our minimization problem. However, for small temporal windows, we can use a simplified assumption, which only models future and past occlusions (one binary variable). Since this is a much simpler problem to solve, it allows a more efficient optimization, which is more likely to find a good solution. Fig. 4.3 visualizes our formulation for $T = 2$ and shows one example for a future and past occlusion. Per definition, all pixels are visible in the reference frame $\bar{\mathbf{I}}_0$. The blue pixel is occluded by the wing of the small dragon in the past. In contrast, the brown pixel is occluded by the wing in the future.

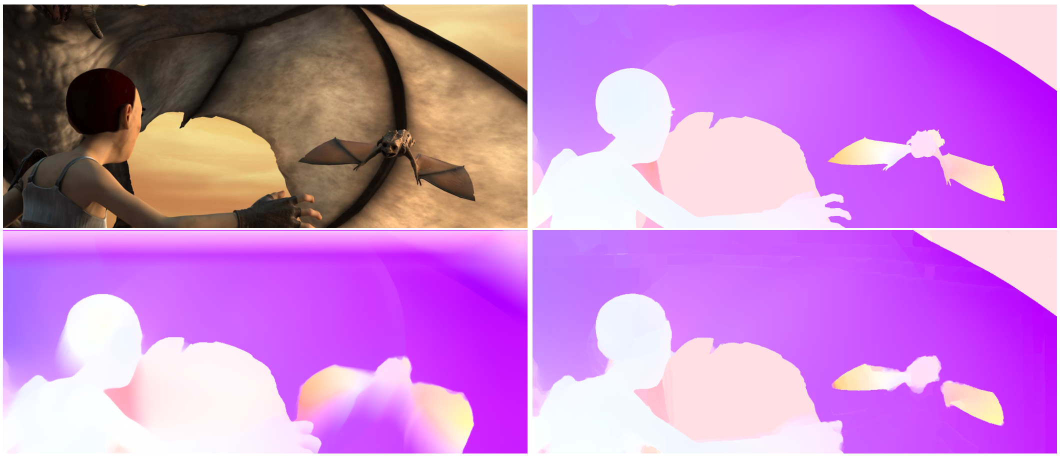


Figure 4.4: **Naive Accumulation.** *We compare the result of a naive accumulation (bottom-left) to our results (bottom-right) and the ground truth (top-right) on the Temple Scene from Sintel (top-left).*

In a short temporal window, we can also expect roughly linear motion because of our high input frame rate. Thus, we enforce constant velocity as a powerful *hard constraint*, as illustrated by the red arrows in Fig. 4.3. While the hard constraint incorporates additional observations for the motion estimation, it does not introduce additional unknown variables and allows for efficient processing of multiple high-resolution input frames. In contrast, a soft constant velocity constraint would introduce $h \cdot w$ variables for each additional frame and could lead to an intractable model.
We formulate the energy functional

$$\mathcal{E}(\mathcal{U},\mathbf{O}) = \int_{\mho} \mathcal{E}^{\mathcal{D}}(\mathcal{U}(\mathbf{p}),\mathbf{O}(\mathbf{p})) + \mathcal{E}^{\mathcal{S}}(\mathcal{U}(\mathbf{p})) + \mathcal{E}^{\mathcal{O}}(\mathbf{O}(\mathbf{p}))\,d\mathbf{p}, \tag{4.1}$$

with $\mathcal{E}^{\mathcal{D}}$ the data term and regularizers $\mathcal{E}^{\mathcal{S}}, \mathcal{E}^{\mathcal{O}}$, which is minimal when $\mathcal{U}, \mathbf{O}$ are the correct optical flow and occlusion mask for a given temporal window.

4.2.1 Data Terms

We design our data term to compare the reference frames with all other frames while taking into account the visibility of each pixel.

Thus, our data term $\mathcal{E}^{\mathcal{D}}$ measures the photo-consistency between the reference frame and future frames if pixel $\mathbf{p}$ is occluded in the past or visible in the entire temporal window ($\mathbf{O}(\mathbf{p}) = 0$). Otherwise, if the pixel is occluded in the future, the photo-consistency between the reference frame and past frames is measured, see Fig. 4.5 for an illustration.

In contrast to a formulation that considers all frames equally in a symmetric window ("Symmetric") or future direction ("Future"), the joint occlusion reasoning allows us to focus only on relevant information for the motion estimation task. Without occlusion reasoning, dominant motions (such as the foreground motion of the finger) affect the estimation in

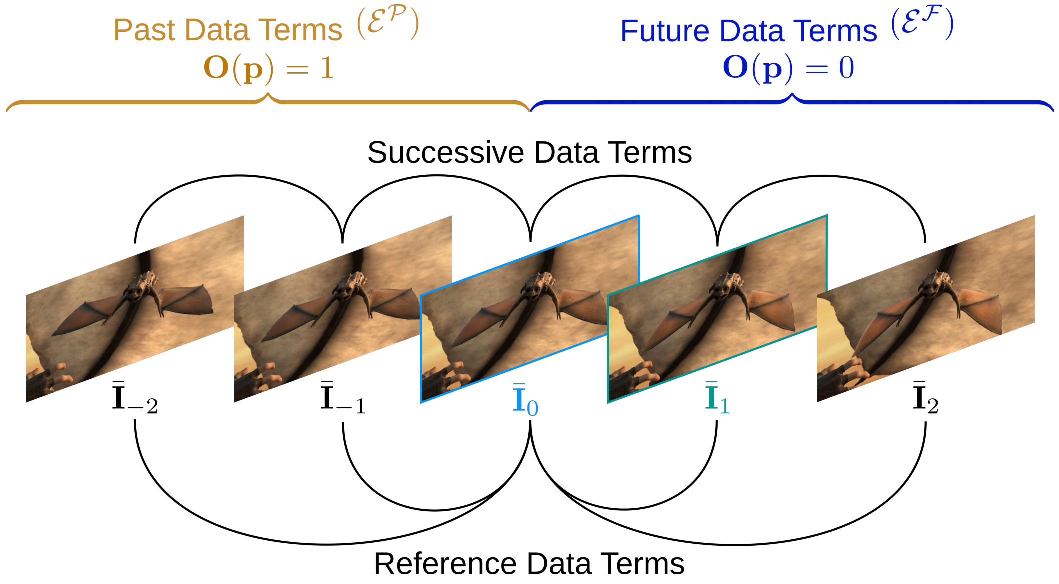


Figure 4.5: **Flowlets Data Terms.** *Illustration of the successive and reference data terms.*

occluded regions resulting in blurring artifacts at motion discontinuities, as illustrated in Fig. 4.6. The joint occlusion reasoning obtains sharp boundaries and small errors even in occluded regions.

We define the data term as

$$\mathcal{E}^{\mathcal{D}}\left(\mathcal{U}\left(\mathbf{p}\right),\mathbf{O}\left(\mathbf{p}\right)\right) = \begin{cases} \mathcal{E}^{\mathcal{F}}\left(\mathcal{U}\left(\mathbf{p}\right)\right) - \mu_{OP} & \text{if } \mathbf{O}\left(\mathbf{p}\right) = 0 \\ \mathcal{E}^{\mathcal{P}}\left(\mathcal{U}\left(\mathbf{p}\right)\right) & \text{otherwise} \end{cases} \tag{4.2}$$

where the bias term μ_{OP} favors future data terms in case neither future nor past occlusions occur.

The future and past photo-consistency terms illustrated in Fig. 4.5 are defined as

$$\mathcal{E}^{\mathcal{F}}\left(\mathcal{U}\left(\mathbf{p}\right)\right) = \mu_A \sum_{t=0}^{T-1} \mathcal{E}_t^{\mathcal{A}}\left(\mathcal{U}\left(\mathbf{p}\right)\right) + \mu_R \sum_{t=1}^{T} \mathcal{E}^{\mathcal{R}}\left(\mathcal{U}\left(\mathbf{p}\right)\right) \tag{4.3}$$

$$\mathcal{E}^{\mathcal{P}}\left(\mathcal{U}\left(\mathbf{p}\right)\right) = \mu_A \sum_{t=-T}^{-1} \mathcal{E}_t^{\mathcal{A}}\left(\mathcal{U}\left(\mathbf{p}\right)\right) + \mu_R \sum_{t=-T}^{-1} \mathcal{E}^{\mathcal{R}}\left(\mathcal{U}\left(\mathbf{p}\right)\right) \tag{4.4}$$

with weighting factors μ_A, μ_R and measure photo-consistency between adjacent frames

$$\mathcal{E}_t^{\mathcal{A}}(\mathcal{U}(\mathbf{p})) = \rho(\bar{\mathbf{I}}_t(\mathbf{p} + t \cdot \mathcal{U}(\mathbf{p})) - \bar{\mathbf{I}}_{t+1}(\mathbf{p} + (t+1) \cdot \mathcal{U}(\mathbf{p}))) \tag{4.5}$$

and with respect to reference frame $\bar{\mathbf{I}}_0$ [WFW08]:

$$\mathcal{E}_t^{\mathcal{R}}(\mathcal{U}(\mathbf{p})) = \rho(\bar{\mathbf{I}}_t(\mathbf{p} + t \cdot \mathcal{U}(\mathbf{p})) - \bar{\mathbf{I}}_0(\mathbf{p})), \tag{4.6}$$

with $\rho(\cdot)$ denoting a robust ℓ_1 penalty function, which operates on the feature channels of $\bar{\mathbf{I}}$. Both data terms serve a different purpose. While the data term on adjacent frames allows for small appearance changes over time, the data term wrt. to the reference frame reduces the drift.

4.2.2 Normalization of Data Terms

The data terms introduced in Eq. (4.5) and Eq. (4.6) are highly non-linear in $\bar{\mathbf{I}}$ and need to be linearized to optimize Eq. (4.1) using the calculus of variations as described in Section 4.2.4. Towards this goal, the first-order Taylor approximation is used. However, Simoncelli, Adelson, and Heeger [SAH91] and Lai and Vemuri [LV98] show that this approximation leads to a weighting of the data term according to the image gradient. This results in high weights when the linear assumption is violated, and they propose to use a normalization of the data term to alleviate this problem. The normalization terms of Eq. (4.5) and Eq. (4.6) are derived as

$$\theta_t^{\mathcal{A}}(\mathbf{p},i) = \left(\left\|t\cdot\nabla\bar{\mathbf{I}}_t(\mathbf{p}+t\cdot\mathcal{U}(\mathbf{p}),i)-(t+1)\cdot\nabla\bar{\mathbf{I}}_{t+1}(\mathbf{p}+(t+1)\cdot\mathcal{U}(\mathbf{p}),i)\right\|_2^2+\varepsilon^2\right)^{-1}$$
$$\theta_t^{\mathcal{R}}(\mathbf{p},i) = \left(t^2\cdot\left\|\nabla\bar{\mathbf{I}}_t(\mathbf{p}+t\cdot\mathcal{U}(\mathbf{p}),i)\right\|_2^2+\varepsilon^2\right)^{-1}$$

with $\theta^{\mathcal{R}}$, $\theta^{\mathcal{A}}$ denoting vectors of dimension c, $\theta_t^{\mathcal{R}}(\mathbf{p},i)$ the i'th column of the vector $\theta_t^{\mathcal{R}}(\mathbf{p})$, $\bar{\mathbf{I}}_t(\mathbf{p},i)$ the i'th channel of frame $\bar{\mathbf{I}}_t(\mathbf{p})$. $\nabla_x\bar{\mathbf{I}}_t(x,y,i)=\bar{\mathbf{I}}_t(x,y,i)-\bar{\mathbf{I}}_t(x-1,y,i)$ denotes the forward difference in direction x and $\varepsilon = 0.001$ is a small constant to prevent the amplification of errors with small gradients. Using these normalization factors, we obtain the normalized data terms

$$\mathcal{E}_t^{\mathcal{A}}(\mathcal{U}(\mathbf{p})) = \rho\left(\theta_t^{\mathcal{A}}(\mathbf{p})\odot\left(\bar{\mathbf{I}}_t(\mathbf{p}+t\cdot\mathcal{U}(\mathbf{p}))-\bar{\mathbf{I}}_{t+1}(\mathbf{p}+(t+1)\cdot\mathcal{U}(\mathbf{p}))\right)\right) \tag{4.7}$$
$$\mathcal{E}_t^{\mathcal{R}}(\mathcal{U}(\mathbf{p})) = \rho\left(\theta_t^{\mathcal{R}}(\mathbf{p})\odot\left(\bar{\mathbf{I}}_t(\mathbf{p}+t\cdot\mathcal{U}(\mathbf{p}))-\bar{\mathbf{I}}_0(\mathbf{p})\right)\right), \tag{4.8}$$

with $\odot$ the element-wise multiplication.

4.2.3 Regularization

Even though our data terms incorporate additional temporal observations, they will not resolve all ambiguities, for example, such caused by untextured regions or occlusions. This will affect both our flow and occlusions variables. Therefore, we impose additional spatial smoothness constraints on the flow ($\mathcal{E}^{\mathcal{S}}$) and occlusion variables ($\mathcal{E}^{\mathcal{O}}$):

$$\mathcal{E}^{\mathcal{S}}(\mathcal{U}(\mathbf{p})) = \mu_{FS}\exp(-\kappa\|\nabla\bar{\mathbf{I}}_0(\mathbf{p})\|_2)\cdot\rho(\|\nabla\mathcal{U}(\mathbf{p})\|_2^2) \tag{4.9}$$
$$\mathcal{E}^{\mathcal{O}}(\mathbf{O}(\mathbf{p})) = \mu_{OS}\|\nabla\mathbf{O}(\mathbf{p})\|_2 \tag{4.10}$$

The smoothness constraints encourage similar flow and occlusion variables between

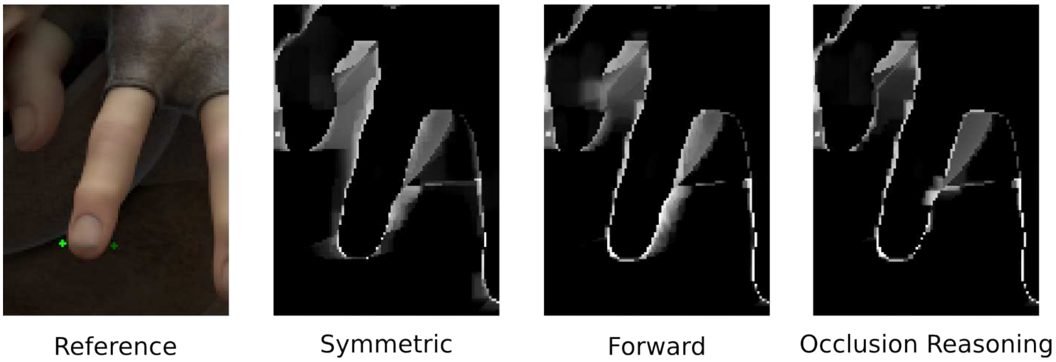


Figure 4.6: **Results with Occlusion Reasoning.** *Visualization of the Average End-point Error (larger errors in brighter colors) using a symmetric data term ($\mathcal{E}^{\mathcal{D}} = \mathcal{E}^{\mathcal{F}} + \mathcal{E}^{\mathcal{P}}$), future photo-consistency ($\mathcal{E}^{\mathcal{D}} = \mathcal{E}^{\mathcal{F}}$) and our full model ($\mathcal{E}^{\mathcal{D}}$ as defined in Eq.* (4.2)*).*

neighboring pixels. In the case of ambiguities, these constraints allow propagating information from neighboring regions. However, the weighting factor $\kappa = 10$ in Eq. (4.9) encourages flow discontinuities at image edges.

4.2.4 Optimization

We minimize Eq. (4.1) by interleaving variational optimization [Bro+04] of the continuous flow variables $\mathcal{U}$ with MAP inference [BVZ99] of the discrete variables $\mathbf{O}$. Depending on the scene and frame rate, we might have larger motions than one pixel. Instead of relying on a scale pyramid (coarse-to-fine) to avoid local minima, we use sparse matching between the reference frame 0 and frame T in combination with the interpolation scheme from EpicFlow [Rev+15] to obtain a good initialization for the optical flow. The alternating variational and discrete optimization yield highly accurate flow fields for small displacements, which form the input to our dense pixel tracking stage.

Discrete Optimization

During the discrete optimization of the occlusion variables, we keep the optical flow $\mathcal{U}$ fixed and minimize $\mathcal{E}(\mathbf{O})$. Since we model only two occlusion states (past or future occlusion), minimizing our energy reduces to a binary optimization problem of the data term $\mathcal{E}^{\mathcal{D}}(\mathbf{O}(\mathbf{p}))$ and regularization $\mathcal{E}^{\mathcal{O}}(\mathbf{O}(\mathbf{p}))$. As discussed in Section 2.2.1, graph cuts approaches are guaranteed to reach the global optimum for such discrete optimization problems. Therefore, we use graph cuts in each iteration to find the optimal solution of our occlusion variables for the current energy. Since we consider the future and past frame, our data terms will always provide information about the appearance of the pixel. In contrast to a two frame formulation, we do not need to account for the trivial solution where all pixels are occluded and the data term is not providing any information.

Continuous Optimization

We assume that our occlusion variables are fixed and find the minimum of our functional $\mathcal{E}(\mathcal{U})$ using the Euler-Lagrange equation as discussed in Section 2.1.

We follow the same strategy as [Bro+04] and use the non-linear data term proposed in Section 4.2.1. For better readability, we define

$$\bar{\mathbf{I}}_t = \bar{\mathbf{I}}_t\left(\mathbf{p} + t \cdot \mathcal{U}(\mathbf{p})\right), \tag{4.11}$$

$$\bar{\mathbf{I}}^A = \bar{\mathbf{I}}_t - \bar{\mathbf{I}}_{t+1}, \tag{4.12}$$

$$\bar{\mathbf{I}}^R = \bar{\mathbf{I}}_t - \bar{\mathbf{I}}_0, \tag{4.13}$$

$$\theta^{\mathcal{A}} = \theta_t^{\mathcal{A}}(\mathbf{p}), \tag{4.14}$$

$$\theta^{\mathcal{R}} = \theta_t^{\mathcal{R}}(\mathbf{p}), \tag{4.15}$$

and obtain the following Euler-Lagrange equation for Eq. (4.1)

$$\begin{aligned}
0 =& \mu_A \cdot \mathbf{O}(\mathbf{p}) \sum_{t=-T}^{-1} \rho'\left(\theta^{\mathcal{A}} \odot \bar{\mathbf{I}}^A\right) \cdot \left(\mathbf{J}_{\bar{\mathbf{I}}_t}^T \cdot t - \mathbf{J}_{\bar{\mathbf{I}}_{t+1}}^T \cdot (t+1)\right) \cdot \left(\theta^{\mathcal{A}} \odot \bar{\mathbf{I}}^A\right) \\
&+ \mu_A \cdot (1 - \mathbf{O}(\mathbf{p})) \sum_{t=0}^{T-1} \rho'\left(\theta^{\mathcal{A}} \odot \bar{\mathbf{I}}^A\right) \cdot \left(\mathbf{J}_{\bar{\mathbf{I}}_t}^T \cdot t - \mathbf{J}_{\bar{\mathbf{I}}_{t+1}}^T \cdot (t+1)\right) \cdot \left(\theta^{\mathcal{A}} \odot \bar{\mathbf{I}}^A\right) \\
&+ \mu_R \cdot \mathbf{O}(\mathbf{p}) \sum_{t=-T}^{-1} \rho'\left(\theta^{\mathcal{R}} \odot \bar{\mathbf{I}}^R\right) \cdot \mathbf{J}_{\bar{\mathbf{I}}_t}^T \cdot t \cdot \left(\theta^{\mathcal{R}} \odot \bar{\mathbf{I}}^R\right) \\
&+ \mu_R \cdot (1 - \mathbf{O}(\mathbf{p})) \sum_{t=1}^{T} \rho'\left(\theta^{\mathcal{R}} \odot \bar{\mathbf{I}}^R\right) \cdot \mathbf{J}_{\bar{\mathbf{I}}_t}^T \cdot t \cdot \left(\theta^{\mathcal{R}} \odot \bar{\mathbf{I}}^R\right) \\
&- \alpha \cdot \exp\left(-\kappa \|\nabla \bar{\mathbf{I}}_0(\mathbf{p})\|_2\right) \cdot \operatorname{div}\left(\rho'\left(\|\nabla \mathcal{U}\|_2^2\right) \cdot \nabla \mathcal{U}\right),
\end{aligned} \tag{4.16}$$

with $\mathbf{J}_{\bar{\mathbf{I}}_t}$ the Jacobian of $\bar{\mathbf{I}}_t$.

We handle the non-linearities in $\mathcal{U}$ with the same numerical approximation as Brox et al. and use fixed-point iterations combined with a scale pyramid to avoid local minima (coarse-to-fine). However, we rely on an EpicFlow initialization u_0 to start on a finer scale than Brox et al. and use the variational optimization for refinement. Denoting $\mathbf{u}^k$, $k = 0, 1, \dots, N$ the estimate of $\mathcal{U}(\mathbf{p})$ at pixel $\mathbf{p}$ in iteration k, we rewrite the equation system as follows

$$\bar{\mathbf{I}}_t^k = \bar{\mathbf{I}}_t\left(\mathbf{p} + t \cdot \mathbf{u}^k\right), \tag{4.17}$$

$$\bar{\mathbf{I}}^{A,k} = \bar{\mathbf{I}}_t^k - \bar{\mathbf{I}}_{t+1}^k, \tag{4.18}$$

$$\bar{\mathbf{I}}^{R,k} = \bar{\mathbf{I}}_t^k - \bar{\mathbf{I}}_0^k, \tag{4.19}$$

$$\theta^{\mathcal{A},k} = \left(\left\|t \cdot \nabla \bar{\mathbf{I}}_t^k - (t+1) \cdot \nabla \bar{\mathbf{I}}_{t+1}^k\right\|_2^2 + \varepsilon^2\right)^{-1} \tag{4.20}$$

$$\theta^{\mathcal{R},k} = \left(t^2 \cdot \left\|\nabla \bar{\mathbf{I}}_t^k\right\|_2^2 + \varepsilon^2\right)^{-1} \tag{4.21}$$

In each fixed-point iteration $k+1$, we use the approximation of the Jacobians $\mathbf{J}_{\bar{\mathbf{I}}_t^k}^T$ and normalization factors $\theta^{\mathcal{A},k}$ and $\theta^{\mathcal{R},k}$ from the previous iteration k.

$$\begin{aligned}
0 =& \mu_A \cdot \mathbf{O}(\mathbf{p}) \sum_{t=-T}^{-1} \rho'\left(\theta^{\mathcal{A},k} \odot \bar{\mathbf{I}}^{A,k+1}\right) \cdot \left(\mathbf{J}_{\bar{\mathbf{I}}_t^k}^T \cdot t - \mathbf{J}_{\bar{\mathbf{I}}_{t+1}^k}^T \cdot (t+1)\right) \cdot \theta^{\mathcal{A},k} \odot \bar{\mathbf{I}}^{A,k+1} \\
&+ \mu_A \cdot (1-\mathbf{O}(\mathbf{p})) \sum_{t=0}^{T-1} \rho'\left(\theta^{\mathcal{A},k} \odot \bar{\mathbf{I}}^{A,k+1}\right) \cdot \left(\mathbf{J}_{\bar{\mathbf{I}}_t^k}^T \cdot t - \mathbf{J}_{\bar{\mathbf{I}}_{t+1}^k}^T \cdot (t+1)\right) \cdot \theta^{\mathcal{A},k} \odot \bar{\mathbf{I}}^{A,k+1} \\
&+ \mu_R \cdot \mathbf{O}(\mathbf{p}) \sum_{t=-T}^{-1} \rho'\left(\theta^{\mathcal{R},k} \odot \bar{\mathbf{I}}^{R,k+1}\right) \cdot \mathbf{J}_{\bar{\mathbf{I}}_t^k}^T \cdot t \cdot \theta^{\mathcal{R},k} \odot \bar{\mathbf{I}}^{R,k+1} \\
&+ \mu_R \cdot (1-\mathbf{O}(\mathbf{p})) \sum_{t=1}^{T} \rho'\left(\theta^{\mathcal{R},k} \odot \bar{\mathbf{I}}^{R,k+1}\right) \cdot \mathbf{J}_{\bar{\mathbf{I}}_t^k}^T \cdot t \cdot \theta^{\mathcal{R},k} \odot \bar{\mathbf{I}}^{R,k+1} \\
&- \alpha \cdot \exp\left(-\kappa \|\nabla \bar{\mathbf{I}}_0(\mathbf{p})\|_2\right) \cdot \operatorname{div}\left(\rho'\left(\|\nabla \mathbf{u}^{k+1}\|_2^2\right) \cdot \nabla \mathbf{u}^{k+1}\right). \qquad (4.22)
\end{aligned}$$

With the first-order Taylor expansion, we handle the non-linearities in the images $\bar{\mathbf{I}}_t$ and now optimize over $d\mathbf{u}^k$. Therefore, we insert the following equations

$$\mathbf{u}^{k+1} = \mathbf{u}^k + d\mathbf{u}^k, \qquad (4.23)$$

$$\bar{\mathbf{I}}_t^{k+1} \approx \bar{\mathbf{I}}_t^k + \mathbf{J}_{\bar{\mathbf{I}}_t^k} \cdot d\mathbf{u}^k, \qquad (4.24)$$

into Eq. (4.22) and get

$$\begin{aligned}
0 =& \mu_A \cdot \mathbf{O}(\mathbf{p}) \sum_{t=-T}^{-1} \Big[\rho'\left(\theta^{\mathcal{A},k} \odot \left(\bar{\mathbf{I}}^{A,k} + \mathbf{J}_{\bar{\mathbf{I}}_t^k} \cdot t \cdot d\mathbf{u}^k - \mathbf{J}_{\bar{\mathbf{I}}_{t+1}^k} \cdot (t+1) \cdot d\mathbf{u}^k\right)\right) \\
&\quad \cdot \left(\mathbf{J}_{\bar{\mathbf{I}}_t^k}^T \cdot t - \mathbf{J}_{\bar{\mathbf{I}}_{t+1}^k}^T \cdot (t+1)\right) \cdot \theta^{\mathcal{A},k} \odot \left(\bar{\mathbf{I}}^{A,k} + \mathbf{J}_{\bar{\mathbf{I}}_t^k} \cdot t \cdot d\mathbf{u}^k - \mathbf{J}_{\bar{\mathbf{I}}_{t+1}^k} \cdot (t+1) \cdot d\mathbf{u}^k\right)\Big] \\
&+ \mu_A \cdot (1-\mathbf{O}(\mathbf{p})) \sum_{t=0}^{T-1} \mu_A \Big[\rho'\left(\theta^{\mathcal{A},k} \odot \left(\bar{\mathbf{I}}^{A,k} + \mathbf{J}_{\bar{\mathbf{I}}_t^k} \cdot t \cdot d\mathbf{u}^k - \mathbf{J}_{\bar{\mathbf{I}}_{t+1}^k} \cdot (t+1) \cdot d\mathbf{u}^k\right)\right) \\
&\quad \cdot \left(\mathbf{J}_{\bar{\mathbf{I}}_t^k}^T \cdot t - \mathbf{J}_{\bar{\mathbf{I}}_{t+1}^k}^T \cdot (t+1)\right) \cdot \theta^{\mathcal{A},k} \odot \left(\bar{\mathbf{I}}^{A,k} + \mathbf{J}_{\bar{\mathbf{I}}_t^k} \cdot t \cdot d\mathbf{u}^k - \mathbf{J}_{\bar{\mathbf{I}}_{t+1}^k} \cdot (t+1) \cdot d\mathbf{u}^k\right)\Big] \\
&+ \mu_R \cdot \mathbf{O}(\mathbf{p}) \sum_{t=-T}^{-1} \rho'\left(\theta^{\mathcal{R},k} \odot \left(\bar{\mathbf{I}}^{R,k} + \mathbf{J}_{\bar{\mathbf{I}}_t^k} \cdot t \cdot d\mathbf{u}^k\right)\right) \cdot \mathbf{J}_{\bar{\mathbf{I}}_t^k}^T \cdot t \cdot \theta^{\mathcal{R},k} \odot \left(\bar{\mathbf{I}}^{R,k} + \mathbf{J}_{\bar{\mathbf{I}}_t^k} \cdot t \cdot d\mathbf{u}^k\right) \\
&+ \mu_R \cdot (1-\mathbf{O}(\mathbf{p})) \sum_{t=1}^{T} \rho'\left(\theta^{\mathcal{R},k} \odot \left(\bar{\mathbf{I}}^{R,k} + \mathbf{J}_{\bar{\mathbf{I}}_t^k} \cdot t \cdot d\mathbf{u}^k\right)\right) \cdot \mathbf{J}_{\bar{\mathbf{I}}_t^k}^T \cdot t \cdot \theta^{\mathcal{R},k} \odot \left(\bar{\mathbf{I}}^{R,k} + \mathbf{J}_{\bar{\mathbf{I}}_t^k} \cdot t \cdot d\mathbf{u}^k\right) \\
&- \alpha \cdot \exp\left(-\kappa \|\nabla \bar{\mathbf{I}}_0(\mathbf{p})\|_2\right) \cdot \operatorname{div}\left(\rho'\left(\|\nabla\left(\mathbf{u}^k + d\mathbf{u}^k\right)\|_2^2\right) \cdot \nabla\left(\mathbf{u}^k + d\mathbf{u}^k\right)\right). \qquad (4.25)
\end{aligned}$$

Finally, we introduce inner fixed-point iterations over $d\mathbf{u}^{k,l}$ for the non-linearities in the robust function ρ, and solve the resulting equation system using Successive Over-Relaxation (SOR). We use bilinear interpolation for evaluating $\bar{\mathbf{I}}$.

4.3 Dense Tracking

Given the Flowlets $\{\mathcal{U}_{t\to t+1}\}$ from the previous section, our goal is to estimate the final optical flow field $\mathbf{U}_{1\to N}$ from frame 1 to frame N. In the following, we formulate the problem as a dense pixel tracking task.

Let $\mathcal{H} = \{\mathbf{H}_1,\dots,\mathbf{H}_N\}$ and $\mathcal{V} = \{\mathbf{V}_1,\dots,\mathbf{V}_N\}$ denote the location and visibility state of each pixel of reference image $\mathbf{I}_1$ in each frame of the full sequence. The domain of a pixel $\mathbf{p} \in \Omega$ in image $\mathbf{I}_t$ is defined as $\Omega = \{1,\dots,w\} \times \{1,\dots,h\}$. Instead of the optical flow, $\mathbf{H}_t \in \mathbb{R}^{w\times h\times 2}$ describes a *location field* and $\mathbf{H}_1$ comprises the location of each pixel in the reference image. Thus, we obtain the optical flow from frame 1 to frame N by $\mathbf{U}_{1\to N} = \mathbf{H}_N - \mathbf{H}_1$. In addition, $\mathbf{V}_t \in \{0,1\}^{w\times h}$ is a *visibility field* (1="visible", 0="occluded"), and by definition, all pixels are visible in the reference frame, $\mathbf{V}_1 = \mathbf{1}^{w\times h}$. The trajectory and visibility variables along the trajectory of each pixel $\mathbf{p} \in \Omega$ in reference image $\mathbf{I}_1$ from frame 1 to frame N are represented by $\mathbf{h}_\mathbf{p} = \{\mathbf{H}_1(\mathbf{p}),\dots,\mathbf{H}_N(\mathbf{p})\}$ and $\mathbf{v}_\mathbf{p} = \{\mathbf{V}_1(\mathbf{p}),\dots,\mathbf{V}_N(\mathbf{p})\}$.

Our goal is to jointly estimate dense pixel trajectories $\mathcal{H}_* = \mathcal{H}\backslash\mathbf{H}_1$ and the visibility label of each point in each frame $\mathcal{V}_* = \mathcal{V}\backslash\mathbf{V}_1$. Again, we cast this task as an energy minimization problem

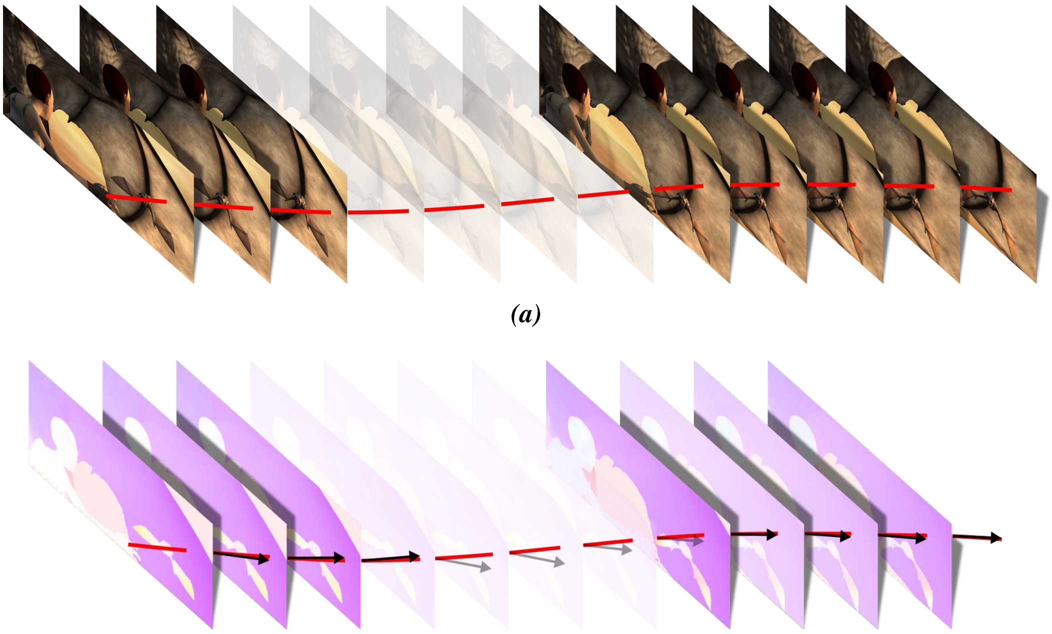
$$\begin{aligned}\mathcal{E}_T(\mathcal{H}_*,\mathcal{V}_*) =& \lambda_{\mathcal{D}_A}\sum_{t<s}\underbrace{\psi_{ts}^{\mathcal{D}_A}(\mathbf{H}_t,\mathbf{V}_t,\mathbf{H}_s,\mathbf{V}_s)}_{\text{Appearance Data Term}}\\ &+\lambda_{\mathcal{D}_F}\sum_{s=t+1}\underbrace{\psi_{ts}^{\mathcal{D}_F}(\mathbf{H}_t,\mathbf{V}_t,\mathbf{H}_s,\mathbf{V}_s)}_{\text{Flow Data Term}}\\ &+\lambda_{\mathcal{U}_T}\sum_{\mathbf{p}\in\Omega}\underbrace{\psi_{\mathbf{p}}^{\mathcal{U}_T}(\mathbf{h}_\mathbf{p})}_{\text{Temporal Flow}}+\lambda_{\mathcal{U}_S}\sum_{\mathbf{p}\sim\mathbf{q}}\underbrace{\psi_{\mathbf{pq}}^{\mathcal{U}_S}(\mathbf{h}_\mathbf{p},\mathbf{h}_\mathbf{q})}_{\text{Spatial Flow}}\\ &+\lambda_{\mathcal{V}_T}\sum_{\mathbf{p}\in\Omega}\underbrace{\psi_{\mathbf{p}}^{\mathcal{V}_T}(\mathbf{v}_\mathbf{p})}_{\text{Temporal Vis.}}+\lambda_{\mathcal{V}_S}\sum_{\mathbf{p}\sim\mathbf{q}}\underbrace{\psi_{\mathbf{pq}}^{\mathcal{V}_S}(\mathbf{v}_\mathbf{p},\mathbf{v}_\mathbf{q})}_{\text{Spatial Vis.}},\end{aligned}\tag{4.26}$$

where $\psi_{ts}^{\mathcal{D}_A}$, $\psi_{ts}^{\mathcal{D}_F}$, $\psi_{\mathbf{p}}^{\mathcal{U}_T}$, $\psi_{\mathbf{pq}}^{\mathcal{U}_S}$, $\psi_{\mathbf{p}}^{\mathcal{V}_T}$, $\psi_{\mathbf{pq}}^{\mathcal{V}_S}$ are data, smoothness and occlusion constraints, and $\{\lambda\}$ are linear weighting factors. Here, $\mathbf{p}\sim\mathbf{q}$ denotes all neighboring pixels $\mathbf{p}\in\Omega$ and $\mathbf{q}\in\Omega$ on a 4-connected pixel grid.

4.3.1 Data Terms

We use all intermediate frames between 1 and N to enforce a constant appearance (Fig. 4.7a). However, we can now also directly constraint the trajectories using the information from the Flowlets. Each Flowlet describes a part of the trajectory, as illustrated in Fig. 4.7b with the trajectory in red and the Flowlets as black arrows. We use a combination of the appearance and flow data terms to leverage as much information as possible.

The **appearance data term** $\psi_{ts}^{\mathcal{D}_A}$ robustly measures the photo-consistency. In such long sequences, appearance changes are much more likely to occur. Therefore, we measure the photo-consistency between each combination of frame t and frame s at all visible pixels. Towards this goal, the features $\mathbf{I}_t,\mathbf{I}_s$ are warped according to the respective location fields



(a)

(b)

Figure 4.7: **Dense Tracking Data Terms.** *Illustration of (a) the appearance data term and (b) the flow data term from the dense tracking formulation. The trajectory* $\mathbf{h}_\mathbf{p}$ *is illustrated in red while the visibility state* $\mathbf{v}_\mathbf{p}$ *is represented by the transparency of the image and flow field. The flow data term (b) compares the trajectory to the Flowlets* $\mathcal{U}_{t\rightarrow t+1}$ *represented by the color encoding and the black arrows.*

$\mathbf{H}_t$ and $\mathbf{H}_s$.

$$\psi_{ts}^{\mathcal{D}_A}(\mathbf{H}_t,\mathbf{V}_t,\mathbf{H}_s,\mathbf{V}_s) = \sum_{\mathbf{p}\in\Omega} \mathbf{V}_t(\mathbf{p})\,\mathbf{V}_s(\mathbf{p})\,\|\mathbf{I}_t(\mathbf{H}_t(\mathbf{p})) - \mathbf{I}_s(\mathbf{H}_s(\mathbf{p}))\|_1 \tag{4.27}$$

with $\mathbf{V}_t(\mathbf{p}) \in \{0,1\}$ indicating the visibility of pixel $\mathbf{p}$ in frame t. For extracting features at fractional locations $\mathbf{p}'_t$ we use again bilinear interpolation. Similarly to the reference and successive data term introduced in Section 4.2.1, comparing adjacent frames allows small appearance changes while the comparison of remote frames alleviates the drift problem.

The **flow data term** $\psi_{ts}^{\mathcal{D}_F}$ measures the agreement between the predicted location field and the Flowlets:

$$\psi_{ts}^{\mathcal{D}_F}(\mathbf{H}_t,\mathbf{V}_t,\mathbf{H}_s,\mathbf{V}_s) = \sum_{\mathbf{p}\in\Omega} \mathbf{V}_t(\mathbf{p})\,\mathbf{V}_s(\mathbf{p})\,\|\mathbf{H}_s(\mathbf{p}) - \mathbf{H}_t(\mathbf{p}) - \mathcal{U}_{t\rightarrow s}(\mathbf{H}_t(\mathbf{p}))\|_1 \tag{4.28}$$

While the appearance term reduces long-range drift, the flow term helps guide the optimization to the global optimum.

4.3.2 Regularization

While our data terms incorporate many observations from our HFR sequence, we still need additional constraints to resolve ambiguities.

In the case of occlusions, our data terms will not provide any information about the trajectory. The **temporal flow term** $\psi_{\mathbf{p}}^{\mathcal{U}_T}$ and **spatial flow term** $\psi_{\mathbf{pq}}^{\mathcal{U}_S}$ allow propagating information in space and time from neighboring trajectories. Towards this goal, $\psi_{\mathbf{p}}^{\mathcal{U}_T}$ robustly penalizes deviations from the constant velocity assumption and $\psi_{\mathbf{pq}}^{\mathcal{U}_S}$ encourages similar trajectories at reference pixels $\mathbf{p}$ and $\mathbf{q}$

$$\psi_{\mathbf{p}}^{\mathcal{U}_T}(\mathbf{h_p}) = \sum_{t=2}^{N-1} \|\mathbf{h}_{\mathbf{p}}^{t-1} - 2\,\mathbf{h}_{\mathbf{p}}^{t} + \mathbf{h}_{\mathbf{p}}^{t+1}\|_1 \tag{4.29}$$

$$\psi_{\mathbf{pq}}^{\mathcal{U}_S}(\mathbf{h_p}, \mathbf{h_q}) = \xi(\mathbf{p},\mathbf{q}) \sum_{t=2}^{N} \|(\mathbf{h}_{\mathbf{p}}^{t} - \mathbf{h}_{\mathbf{p}}^{t-1}) - (\mathbf{h}_{\mathbf{q}}^{t} - \mathbf{h}_{\mathbf{q}}^{t-1})\|_2 \tag{4.30}$$

with $\mathbf{h}_{\mathbf{p}}^{t}$ the location of reference pixel $\mathbf{p}$ in frame t and $\xi(\mathbf{p},\mathbf{q}) = \exp(-\kappa \|\nabla \mathbf{I}_1(\frac{\mathbf{p}+\mathbf{q}}{2})\|_2)$ with $\kappa = 10$ a weighting factor, which encourages flow discontinuities at image edges.

However, a trivial solution to our energy so far would be to set every pixel occluded. We can resolve this problem by encouraging the visible state. In addition, we can make similar temporal and spatial assumptions on our occlusion variables as on our trajectories. Usually, occlusions are caused by other objects affecting a larger image region instead of single pixels. In addition, occlusions usually last for a certain time period and do not change every frame. Thus, we expect the occlusion variables to change smoothly over time and space. We introduce the **temporal visibility term** $\psi_{\mathbf{p}}^{\mathcal{V}_T}$ that penalizes temporal changes of the visibility of a pixel $\mathbf{p}$ via a Potts model (first part) and encodes our belief that the majority of pixels in each frame should be visible (second part). Finally, the **spatial visibility term** $\psi_{\mathbf{pq}}^{\mathcal{V}_S}$ encourages neighboring trajectories to take on similar visibility labels modulated by the contrast-sensitive smoothness weight ξ.

$$\psi_{\mathbf{p}}^{\mathcal{V}_T}(\mathbf{v_p}) = \sum_{t=1}^{N-1} [\mathbf{v}_{\mathbf{p}}^{t} \neq \mathbf{v}_{\mathbf{p}}^{t+1}] - \lambda_{\mathcal{V}} \sum_{t=1}^{N} \mathbf{v}_{\mathbf{p}}^{t}, \tag{4.31}$$

$$\psi_{\mathbf{pq}}^{\mathcal{V}_S}(\mathbf{v_p}, \mathbf{v_q}) = \xi(\mathbf{p},\mathbf{q}) \sum_{t=1}^{N} [\mathbf{v}_{\mathbf{p}}^{t} \neq \mathbf{v}_{\mathbf{q}}^{t}]. \tag{4.32}$$

Here, $\mathbf{v}_{\mathbf{p}}^{t}$ denotes if pixel pixel $\mathbf{p}$ in frame t is visible or not.

4.3.3 Optimization

Unfortunately, finding a minimizer of Eq. (4.26) is a very difficult problem that does not admit the application of black-box optimizers: First, the number of variables to be estimated is orders of magnitude larger than for classical problems in computer vision. For instance, a sequence of 100 QuadHD images results in more than 1 billion variables to be estimated. Second, our energy comprises discrete and continuous variables, which make optimization hard. Finally, the optimization problem is highly non-convex due to the non-linear dependence on the input images. Thus, gradient descent techniques quickly get trapped in local minima when initialized with constant location fields.

In this section, we introduce several simplifications to make approximate inference in our model tractable. As the choice of these simplifications will crucially affect the quality of the retrieved solutions, we will discuss in-depth each of these choices in the following.

Derivation of MRF

We use Max Product Particle Belief Propagation (MP-PBP), discussed in Section 2.2.1, to make the optimization of our discrete-continuous objective feasible. We iteratively discretize the continuous variables, sample the discrete variables, and perform TRW-S [Kol06] on the resulting discrete MRF. More specifically, we create a discrete set of trajectory and visibility hypotheses $\{(\mathbf{h}_\mathbf{p}^{(1)}, \mathbf{v}_\mathbf{p}^{(1)}), \ldots, (\mathbf{h}_\mathbf{p}^{(M)}, \mathbf{v}_\mathbf{p}^{(M)})\}$ for each pixel $\mathbf{p}$. Estimating $\mathcal{X} = \{x_\mathbf{p} | \mathbf{p} \in \Omega\}$ with $x_\mathbf{p} = (\mathbf{h}_\mathbf{p}, \mathbf{v}_\mathbf{p}) \in \{(\mathbf{h}_\mathbf{p}^{(1)}, \mathbf{v}_\mathbf{p}^{(1)}), \ldots, (\mathbf{h}_\mathbf{p}^{(M)}, \mathbf{v}_\mathbf{p}^{(M)})\}$ can be phrased as inference in a simpler Markov Random Field:

By inserting our definitions of the data and smoothness terms, we obtain

$$
\begin{aligned}
E(\mathcal{H}_*, \mathcal{V}_*) = & \lambda_{\mathcal{D}_A} \sum_{\mathbf{p}\in\Omega} \sum_{t<s} \mathbf{v}_\mathbf{p}^t \, \mathbf{v}_\mathbf{p}^s \, \|\mathbf{I}_t(\mathbf{h}_\mathbf{p}^t) - \mathbf{I}_s(\mathbf{h}_\mathbf{p}^s)\|_1 \\
& + \lambda_{\mathcal{D}_F} \sum_{\mathbf{p}\in\Omega} \sum_{s=t+1} \mathbf{v}_\mathbf{p}^t \, \mathbf{v}_\mathbf{p}^s \, \|\mathbf{h}_\mathbf{p}^s - \mathbf{h}_\mathbf{p}^t - \mathcal{U}_{t\to s}(\mathbf{h}_\mathbf{p}^t)\|_1 \\
& + \lambda_{\mathcal{U}_T} \sum_{\mathbf{p}\in\Omega} \sum_{t=2}^{N-1} \|\mathbf{h}_\mathbf{p}^{t-1} - 2\,\mathbf{h}_\mathbf{p}^t + \mathbf{h}_\mathbf{p}^{t+1}\|_1 \\
& + \lambda_{\mathcal{U}_S} \sum_{\mathbf{p}\sim\mathbf{q}} \xi(\mathbf{p},\mathbf{q}) \sum_{t=2}^{N} \|(\mathbf{h}_\mathbf{p}^t - \mathbf{h}_\mathbf{p}^{t-1}) - (\mathbf{h}_\mathbf{q}^t - \mathbf{h}_\mathbf{q}^{t-1})\|_2 \\
& + \lambda_{\mathcal{V}_T} \sum_{\mathbf{p}\in\Omega} \sum_{t=1}^{N-1} [\mathbf{v}_\mathbf{p}^t \neq \mathbf{v}_\mathbf{p}^{t+1}(\mathbf{p})] - \lambda_{\mathcal{V}} \sum_{t=1}^{N} \mathbf{v}_\mathbf{p}^t + \lambda_{\mathcal{V}_S} \sum_{\mathbf{p}\sim\mathbf{q}} \xi(\mathbf{p},\mathbf{q}) \sum_{t=1}^{N} [\mathbf{v}_\mathbf{p}^t \neq \mathbf{v}_\mathbf{q}^t]
\end{aligned}
$$

Finally, re-arranging the terms yields

$$
\begin{aligned}
E(\mathcal{H}_*, \mathcal{V}_*) = \sum_{\mathbf{p}\in\Omega} \Bigg[& \lambda_{\mathcal{D}_A} \sum_{t<s} \mathbf{v}_\mathbf{p}^t \, \mathbf{v}_\mathbf{p}^s \, \|\mathbf{I}_t(\mathbf{h}_\mathbf{p}^t) - \mathbf{I}_s(\mathbf{h}_\mathbf{p}^s)\|_1 + \lambda_{\mathcal{D}_F} \sum_{s=t+1} \mathbf{v}_\mathbf{p}^t \, \mathbf{v}_\mathbf{p}^s \, \|\mathbf{h}_\mathbf{p}^s - \mathbf{h}_\mathbf{p}^t - \mathcal{U}_{t\to s}(\mathbf{h}_\mathbf{p}^t)\|_1 \\
& + \lambda_{\mathcal{U}_T} \sum_{t=2}^{N-1} \|\mathbf{h}_\mathbf{p}^{t-1}(\mathbf{p}) - 2\,\mathbf{h}_\mathbf{p}^t + \mathbf{h}_\mathbf{p}^{t+1}\|_1 + \lambda_{\mathcal{V}_T} \sum_{t=1}^{N-1} [\mathbf{v}_\mathbf{p}^t \neq \mathbf{v}_\mathbf{p}^{t+1}] - \lambda_{\mathcal{V}} \sum_{t=1}^{N} \mathbf{v}_\mathbf{p}^t \Bigg] \\
+ \sum_{\mathbf{p}\sim\mathbf{q}} \xi(\mathbf{p},\mathbf{q}) & \left[\sum_{t=2}^{N} \lambda_{\mathcal{U}_S} \|(\mathbf{h}_\mathbf{p}^t - \mathbf{h}_\mathbf{p}^{t-1}) - (\mathbf{h}_\mathbf{q}^t - \mathbf{h}_\mathbf{q}^{t-1})\|_2 + \sum_{t=1}^{N} \lambda_{\mathcal{V}_S} [\mathbf{v}_\mathbf{p}^t \neq \mathbf{v}_\mathbf{q}^t] \right],
\end{aligned}
$$

which can be written as a unary (ψ^U) and pairwise term (ψ^P)

$$E(\mathcal{X}) = \sum_{\mathbf{p}} \psi^U_{\mathbf{p}}(x_{\mathbf{p}}) + \sum_{\mathbf{p}\sim\mathbf{q}} \psi^P_{\mathbf{pq}}(x_{\mathbf{p}}, x_{\mathbf{q}}) \tag{4.33}$$

where $x_{\mathbf{p}} = (\mathbf{h}_{\mathbf{p}}, \mathbf{v}_{\mathbf{p}})$

$$\begin{aligned}
\psi^U_{\mathbf{p}}(x_{\mathbf{p}}) =& \lambda_{\mathcal{D}_A} \sum_{t<s} \mathbf{v}^t_{\mathbf{p}} \mathbf{v}^s_{\mathbf{p}} \left\|\mathbf{I}_t(\mathbf{h}^t_{\mathbf{p}}) - \mathbf{I}_s(\mathbf{h}^s_{\mathbf{p}})\right\|_1 + \lambda_{\mathcal{D}_F} \sum_{s=t+1} \mathbf{v}^t_{\mathbf{p}} \mathbf{v}^s_{\mathbf{p}} \left\|\mathbf{h}^s_{\mathbf{p}} - \mathbf{h}^t_{\mathbf{p}} - \mathcal{U}_{t\to s}(\mathbf{h}^t_{\mathbf{p}})\right\|_1 \\
&+ \lambda_{\mathcal{U}_T} \sum_{t=2}^{N-1} \left\|\mathbf{h}^{t-1}_{\mathbf{p}}(\mathbf{p}) - 2\mathbf{h}^t_{\mathbf{p}} + \mathbf{h}^{t+1}_{\mathbf{p}}\right\|_1 + \lambda_{\mathcal{V}_T} \sum_{t=1}^{N-1} [\mathbf{v}^t_{\mathbf{p}} \neq \mathbf{v}^{t+1}_{\mathbf{p}}] - \lambda_{\mathcal{V}} \sum_{t=1}^{N} \mathbf{v}^t_{\mathbf{p}} \\
\psi^P_{\mathbf{pq}}(x_{\mathbf{p}}, x_{\mathbf{q}}) =& \xi(\mathbf{p}, \mathbf{q}) \left[\sum_{t=2}^{N} \lambda_{\mathcal{U}_S} \left\|(\mathbf{h}^t_{\mathbf{p}} - \mathbf{h}^{t-1}_{\mathbf{p}}) - (\mathbf{h}^t_{\mathbf{q}} - \mathbf{h}^{t-1}_{\mathbf{q}})\right\|_2 + \sum_{t=1}^{N} \lambda_{\mathcal{V}_S} [\mathbf{v}^t_{\mathbf{p}} \neq \mathbf{v}^t_{\mathbf{q}}]\right]
\end{aligned}$$

Given this discrete set, the optimization of Eq. (4.26) is equivalent to the MAP solution of the simpler MRF given in Eq. (4.33).

Hypothesis Generation

A common strategy for MP-PBP [TM09; GG15] is to start from a random initialization and to generate particles by iteratively resampling from a Gaussian distribution centered at the last MAP solution. This implements a stochastic gradient descent procedure without the need for computing gradients. Unfortunately, our objective is highly non-convex, and random or constant initialization will guide the optimizer to a bad local minimum close to the initialization.

We, therefore, opt for a data-driven hypothesis generation strategy. We accumulate the precomputed Flowlets between all subsequent frames of the input video sequence in temporal direction (forward and backward). As not all pixels are visible during the entire sequence, we detect temporal occlusion boundaries using a forward-backward consistency check and track through partially occluded regions with spatial and temporal extrapolation. We use EpicFlow [Rev+15] to spatially extrapolate the consistent parts of each Flowlet, which allows propagating the flow from the visible into occluded regions. For temporal extrapolation, we predict point trajectories linearly from the last visible segment of each partially occluded trajectory. This strategy works well in cases where the camera and objects move smoothly (e.g., on Sintel or recordings using a tripod) while the temporal linearity assumption is often violated for hand-held recordings. However, spatial extrapolation is usually able to establish correct hypotheses in those cases.

After each run of TRW-S, we resample the particles by sampling hypotheses from spatially neighboring pixels. This allows the propagation of high-quality motions into partial occlusions. In practice, we create a nearest neighbor tree based on the consistent accumulations and retrieve the nearest neighbors for p in a certain radius. We leverage non-maximum suppression based on the following similarity criterion between two hypotheses

$\mathbf{h}_1$ and $\mathbf{h}_2$ to encourage diversity amongst hypotheses:

$$Sim(\mathbf{h}_1, \mathbf{h}_2) = \sum_{t=2}^{N} \|(\mathbf{h}_1^t - \mathbf{h}_1^{t-1}) - (\mathbf{h}_2^t - \mathbf{h}_2^{t-1})\|_2 \tag{4.34}$$

Assuming that the motion of occluders and occludees differs in most cases, we set the visibility of a hypothesis by comparing the local motion prediction with the corresponding Flowlet. If the predicted flow differs significantly from the Flowlet estimate for a particular frame, the pixel is likely occluded.

Spatial Resolution

While a high (QuadHD) input resolution is important to capture fine details and attain subpixel precision, we decided to produce optical flow reference data at half resolution (1280 × 1024 Pixels), which is still significantly larger than all existing optical flow benchmarks [Bak+11; But+12; GLU12].

While using the original resolution for the data term, we estimate $\mathcal{H}$ and $\mathcal{V}$ directly at the output resolution, yielding a 4 fold reduction in model parameters. Note that we do not lose precision in the optical flow field as we continue evaluating the data term at full resolution. To strengthen the data term, we assume that the flow in a small 3×3 pixel neighborhood of the original resolution is constant, yielding 9 observations for each point $\mathbf{p}$ in Eq. (4.27).

Temporal Resolution

While we observed that a high temporal resolution is important for initialization, our temporal smoothness constraints operate more effectively at a coarser resolution as they are able to regularize over larger temporal windows. Additionally, we observe in our experiments in Section 4.4.2 that it is not possible to choose one optimal frame rate due to the trade-off between local estimation accuracy and drift over time, which agrees with the findings in [LAG05].

Therefore, we use two different frame rates for the hypotheses generation and choose the highest frame rate based on the robust upper 90% quantile of the optical flow magnitude computed at a smaller input resolution with classical techniques [Rev+15]. This allows us to choose a fixed maximum displacement between frames. In practice, we chose the largest frame rate that yields maximal displacements of $\sim$2 pixels and the lowest frame rate that yields maximal displacements of $\sim$8 pixels, which empirically gave the best results. Finally, our dense pixel tracking algorithm operates on keyframes based on the lowest frame rate. Flowlet observations of larger frame rates are integrated by accumulating the optical flow between keyframes.

4.4 Evaluation & Analysis

Before we use our method to create reference flow fields for challenging sequences, we first validate our approach by quantifying the error of the reference fields on synthetic and real data with ground truth. All of our real-world sequences are captured with a Fastec TS5Q camera [4], which records QuadHD (2560 × 1440) videos with up to 360 fps.

We empirically determined the optimal weighting parameters for our formulation. For the Flowlets, we use $\mu_G = 6.0, \mu_A = 1.0, \mu_R = 2.0, \mu_{OP} = 500, \mu_{FS} = 4.0, \mu_{OS} = 0.1$, and for the dense tracking formulation, we use $\mu_G = 10.0, \lambda_{\mathcal{D}_A} = 1.0, \lambda_{\mathcal{D}_F} = 1.0, \lambda_{\mathcal{U}_T} = 0.1, \lambda_{\mathcal{U}_S} = 10.0, \lambda_{\mathcal{V}_T} = 1.0, \lambda_{\mathcal{V}_S} = 0.1, \lambda_{\mathcal{V}} = 1.0$.

In all of our experiments, we consider two standard metrics:

- **Average End-point Error (EPE)** is the average Euclidean distance between the estimated and ground truth flow:

$$\text{EPE} = \frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} \left\| \mathbf{U}_{1\to N}(x,y) - \mathbf{U}^{GT}_{1\to N}(x,y) \right\|_2$$

We separately report the EPE in occluded and visible regions to better analyze the impact of the proposed model components.

- **F1-Score** defined as the harmonic mean of precision and recall for occlusion estimates:

$$precision = \frac{TP}{TP+FP}$$
$$recall = \frac{TP}{TP+FN}$$

		Prediction	
		Occluded	Visible
GT	Occluded	TP	FN
	Visible	FP	TN

$$\text{F1-Score} = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

4.4.1 Datasets

As there exists no publicly available HFR dataset with optical flow ground truth, we created two novel datasets for this purpose.

MPI Sintel: We selected a subset of 19 sequences from the MPI Sintel training set [But+12] and re-rendered them in Blender based on the “clean” pass of Sintel at 1008 frames per second, using a resolution of 2048 × 872 pixels. The image quality and realism are identical to that of the original MPI Sintel training set (“clean” pass), except for objects that include physically simulated deformations, like the main character’s hair (Fig. 4.8) or some clothes, as all physical simulations in Sintel are pre-computed at 24 frames per second. While perfect ground truth flow fields can be obtained in this synthetic setting, the rendered images lack realism and textural details.

Real-World Sequences: We thus recorded a second data set of static real-world scenes using our Fastec TS5Q camera. With the dense reconstruction pipeline consisting of Visu-

[4] `http://www.fastecimaging.com/products/handheld-cameras/ts5`

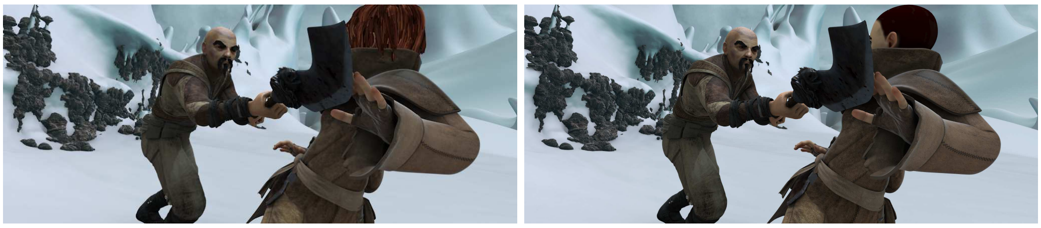


***Figure 4.8: HFR MPI Sintel** We re-rendered the MPI Sintel dataset at 1008fps clean pass to obtain dense accurate ground truth. In this frame rate, we had to remove some effects like the hair of the main character. On the left, we show the original Ambush scene in contrast to the re-rendered version on the right.*

alSFM [Wu13][5] and PMVS2 [FP10][6], we create an accurate 3*D* point cloud from images recorded with the Fastec TS5Q camera. We make sure to obtain a good dense reconstruction by adding a large set of DSLR images (yielding in total 100 - 200 images per reconstruction) to the set of high-speed images and manually deleting points that seem to be wrong. Given the 3*D* point cloud, we can re-project the points into the images and compute the flow between two images. We will be limited to static scenes, and the flow fields will only be sparse in this evaluation. Thus, we will not have such complex motions as in Sintel. However, this experiment will still give us an idea of the performance on our high-speed data since we use the same camera. Our reconstruction dataset consists of 4 point clouds with 20 sequences in total for the evaluation. The point clouds, after manually removing outliers, are shown in Fig. 4.9. In the 20 sequences, we used different camera motions and viewpoints for a diverse dataset.

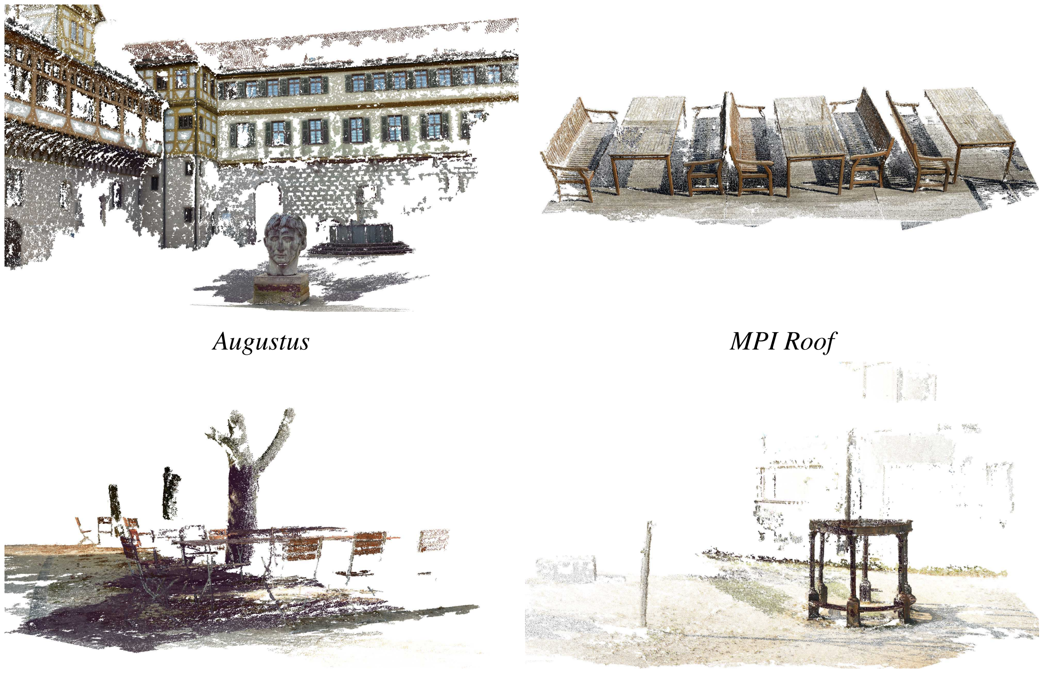
4.4.2 Importance of Frame Rate

Our HFR version of the Sintel dataset allows us to analyze the impact of frame rate on the performance. In the following experiment, we use the naive accumulation of Flowlets on different frame rates to compare them on the original frame rate of 24fps. In addition, we exclude occluded regions since the naive accumulation ignores occlusions. We obtain different frame rates by skipping frames at the highest frame rate of 1008fps. Note that for some frame rates, linear interpolation is necessary to obtain the flow in the original frame rate of 24fps. In the sequences Ambush and Market, the linear interpolation of non-linear motions causes errors that are not entirely following the expected trend.

In Fig. 4.10, we show the estimation error (EPE) with a temporal window size of 3, 5, 9, and 13 frames using different frame rates (x-axis). Overall, we observe decreasing errors with higher frame rates. Interestingly, however, this holds true only until a certain frame rate for a temporal window size of 3 and 5 frames. The reason for the error to increase after the optimal frame rate is the accumulation of small estimation errors that causes a significant drift for higher frame rates. Larger temporal windows perform weaker on lower

[5] A visual Structure-from-Motion system

[6] A visual Multi-View-Stereo system



Augustus

MPI Roof

Sternwarte

Figure 4.9: Reconstruction Dataset *The reconstructed point clouds for 4 different scenes used for evaluating the reference flow fields.*

frame rates but at the same time show a smaller drift over time. Using a larger temporal window can be considered as using a lower frame rate with additional temporal information since we impose the hard constant velocity constraint over a longer time period. Therefore, the optimal frame rate is higher with larger temporal frame rates than with smaller ones, e.g., 144fps using 3 frames and 504 fps using 9 frames temporal windows. The optimal frame rate also highly depends on the scene. Whereas higher frame rates perform better for Ambush, Cave, Market, and Temple, the lowest frame rate of 24fps is optimal for Bamboo and Mountain. In Bamboo and Mountain, we have already very small motions in the original frame rate, thus higher frame rates only lead to a larger drift.

We can therefore conclude as in [LAG05] that while very high frame rates help in general, the optimal frame rate depends not only on the available resources but also on the imaging modalities and the scenario at hand. Thus, it is impossible to choose a single optimal frame rate across all sequences. We therefore use an adaptive frame rate according to the 90% quantile in our dense tracking approach, as described before.

4.4.3 Validation on MPI Sintel

The MPI Sintel data set also allows us to validate our full approach for generating reference data in a low frame rate. Thus given the 1008fps sequence, we use our approach to generate the reference flow field in the original frame rate of 24fps and compare it to the ground truth.

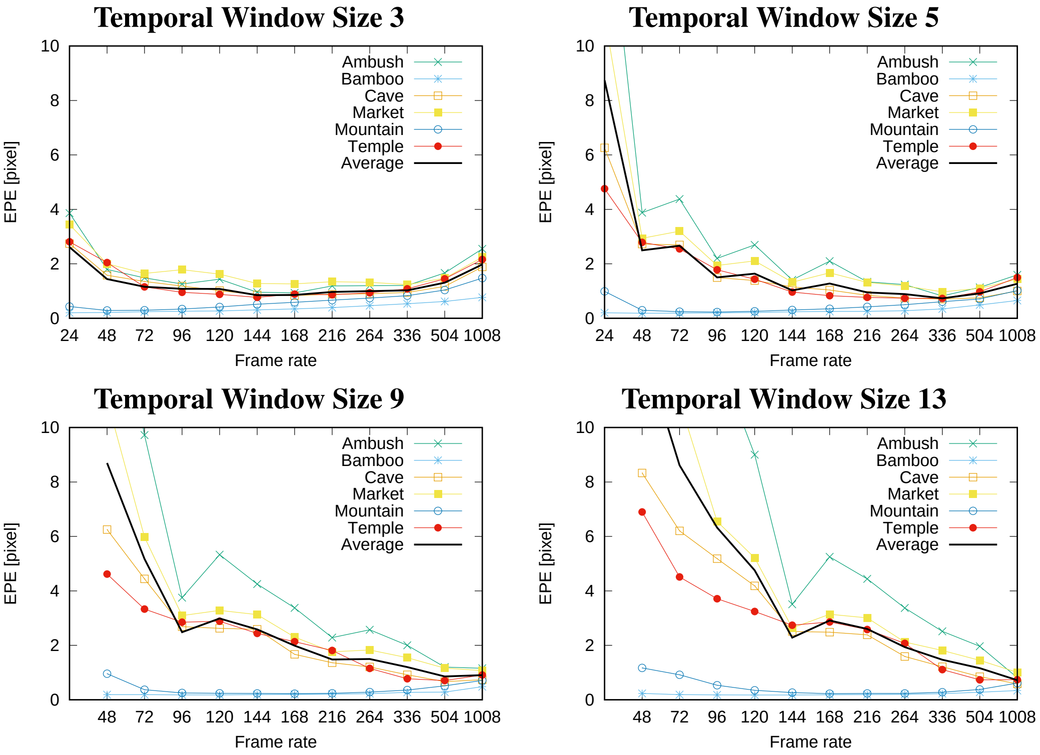


__Figure 4.10: Importance of Frame Rate.__ The EPE on MPI Sintel of non-occluded pixels for different frame rates (x-axis) using temporal window size 3,5,9 and 13 for the Flowlets with naive accumulation.

Table 4.1 shows our results on this dataset evaluated in all regions, only the visible regions, only the occluded regions or regions close to the respective motion boundaries (“Edges”). We also provide the performance on individual scenes in Table 4.2 for a more detailed discussion. We compare our results to Epic Flow [Rev+15] at standard frame rate (24fps), a simple accumulation of EpicFlow flow fields at 144 fps (beyond 144 fps we observed accumulation drift on MPI Sintel), our multi-frame Flowlets (using a windows size of 5) accumulated at the same frame rate and at 1008 fps, as well as our full model.

Compared to computing optical flow at regular frame rates (“Epic Flow (24fps)”), the accumulation of flow fields computed at higher frame rates increases performance in non-occluded regions (“Epic Flow (Accu. 144fps)”). In contrast, occluded regions are not handled by the simple flow accumulation approach and we can observe a small increase in EPE. The proposed multi-frame flow integration (“Slow Flow (Accu. 144fps)”) improves performance further. This is due to our multi-frame data term, which reduces drift during the accumulation. In addition, errors decrease in particular in occluded regions and at motions boundaries (Edges) because of the occlusion reasoning, which considers only the visible frames in case of occlusions. While motion boundaries improve when accumulating multi-frame estimates at higher frame rates (“Slow Flow (Accu. 1008fps)”), the accumulation of

Methods	All (Edges)	Visible (E.)	Occluded (E.)
Epic Flow (24fps)	5.53 (16.23)	2.45 (10.10)	16.54 (20.68)
Epic Flow (Accu. 144fps)	4.73 (12.76)	1.04 (4.41)	17.09 (18.44)
Slow Flow (Accu. 144fps)	4.03 (12.03)	0.78 (4.43)	15.24 (17.28)
Slow Flow (Accu. 1008fps)	5.38 (11.78)	1.35 **(2.60)**	19.18 (17.93)
Slow Flow (Full Model)	**2.40 (10.34)**	**0.75** (4.02)	**9.07 (15.10)**

Table 4.1: **Average Performance on MPI Sintel.** *The performance in EPE of our dense pixel tracking method and various baselines on MPI Sintel with dense ground truth.*

flow errors causes drift resulting in an overall increase in error. This confirms the necessity to choose the frame rate adaptively depending on the expected motion magnitude, as discussed in Section 4.3.3. Using our full model ("Slow Flow (Full Model)"), we obtain the overall best results, reducing errors wrt. EpicFlow at original frame rate by over 60% in visible regions and over 40% in occluded regions.

Table 4.2 provides deeper insights into the performance of our method. Especially in sequences with large and complex motions like "Ambush", "Cave", "Market", and "Temple", we observe significant improvement. We improve in particular in the occluded regions and at motion boundaries due to the propagation of neighboring hypotheses and our occlusion reasoning. However, in scenes like "Bamboo" and "Mountain", the motions are already rather small, which causes a stronger drift problem when considering higher frame rates.

In Table 4.3, we compare the occlusion estimation of our method (last row) to a naive estimate that sets all pixels in the image to occluded (first row) and two-frame EpicFlow in combination with a simple forward-backward check (second row). We report F1-Measure of the estimated occlusion area wrt. the Sintel occlusion ground truth. Our method outperforms both baselines and works best at large occluded regions. The Sintel scenes Bamboo and Mountain comprise very fine occlusions due to small motions that are hard to recover. On Mountain, EpicFlow even outperforms our method in terms of occlusion estimation.

MPI Sintel also contains several easy scenes (e.g., "Bamboo", "Mountain") where state-of-the-optical flow algorithms perform well due to the relatively small motion (around 10 pixel in average). Thus the overall improvement of our method is less pronounced compared to considering the challenging cases alone. However, on more complex scenes with non-rigid

Methods	Ambush	Bamboo	Cave	Market	Mountain	Temple
Epic Flow (24fps)	6.80	0.35	9.07	6.40	1.13	6.84
Epic Flow (Accu. 144fps)	7.09	0.50	5.23	4.88	0.99	6.73
Slow Flow (Accu. 144fps)	5.71	0.32	4.86	4.60	**0.74**	5.46
Slow Flow (Accu. 1008fps)	8.02	0.75	5.55	5.42	1.62	7.88
Slow Flow (Full Model)	**2.85**	**0.25**	**4.76**	**2.37**	0.96	**2.53**

Table 4.2: **Detailed Results on MPI Sintel Scenes.** *The performance in EPE of our dense pixel tracking method and various baselines on several scenes of MPI Sintel.*

Methods	Ambush	Bamboo	Cave	Market	Mountain	Temple	Average
All Occluded	0.40	0.06	0.31	0.27	0.10	0.37	0.28
EpicFlow F/B	0.84	0.27	0.55	0.74	**0.86**	0.77	0.76
Full Model	**0.90**	**0.41**	**0.75**	**0.79**	0.75	**0.85**	**0.82**

Table 4.3: **Occlusions on MPI Sintel Scenes.** *Evaluation of the occlusion estimates of our dense pixel tracking method and various baselines on several scenes from MPI Sintel using the F1-Measure.*

Scene	Augustus			Sternwarte			MPI Roof		
Magnitude	100	200	300	100	200	300	100	200	300
Epic Flow	1.23	5.89	21.63	2.15	5.49	11.03	0.87	23.05	59.36
Slow Flow	**1.17**	**2.56**	**3.71**	**2.04**	**4.99**	**7.59**	**0.86**	**2.01**	**2.70**

Table 4.4: **Validation on Real-World Sequences.** *The accuracy of our dense pixel tracking method and EpicFlow wrt. different motion magnitudes on real-world scenes with ground truth provided by 3D reconstruction.*

objects and larger motions our method always outperforms EpicFlow.

4.4.4 Validation on Real-World Sequences

Using a synthetic data set raises the question of how good the rendered data represent the real world? Instead of answering this question, we also validate our approach on our real-world data set recorded with the Fastec TS5Q comprising of several static scenes. Note that since we obtain our ground truth from 3D reconstruction, we only have sparse annotations and can only consider simple static scenes. However, this should already give us an idea of the performance in real scenes.

In Table 4.4, we compare our approach to an EpicFlow baseline at regular frame rates. Since we have to rely on static and highly textured scenes for a good reconstruction, our baseline already performs very well on small flow magnitudes of $\sim$ 100 pixels on all scenes. However, the advantage of our approach over the baseline becomes clear with larger magnitudes $\sim$ 200 pixels and $\sim$ 300 pixels. Especially on MPI Roof, EpicFlow is failing with objects moving out of the scenes, which results in large errors at the image boundaries. In contrast, our approach still achieves similar performance with 200 and 300px motion magnitudes.

In Fig. 4.11, we show the generated flow fields from our approach for different flow magnitudes, and in Fig. 4.12, we compare our flow fields to the Epic Flow baseline. All flow illustrations are generated using the Middlebury [Bak+11] color scheme, and for the comparison of two flow fields we normalize by the maximum flow of both. For 100px magnitudes, we observed similar performance of our approach and Epic Flow. Therefore we omitted these flow fields here. In the case of occlusions, Epic Flow is having trouble to estimate the correct flow, whereas our motion boundaries are much better. For instance, the large occlusions by the statue in the first and second row. At the same time, we observe

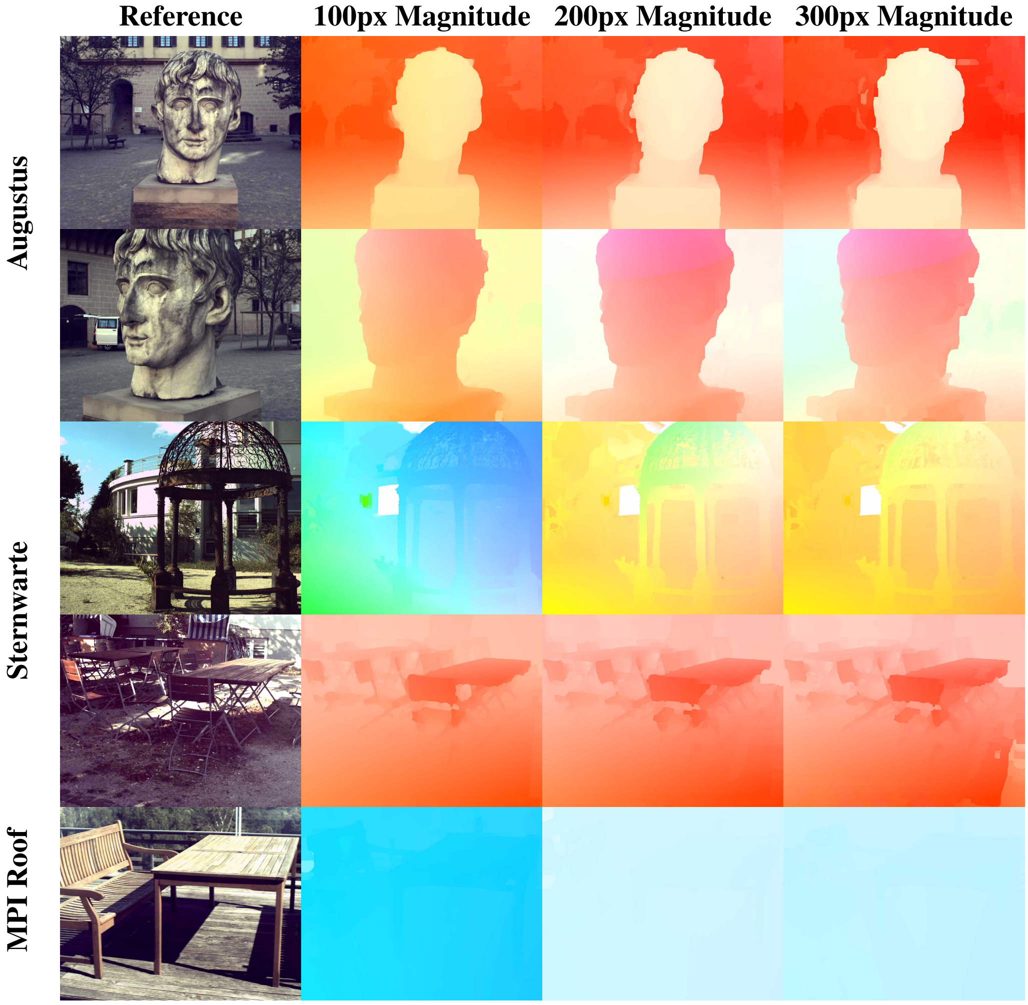


Figure 4.11: **Estimation on Real-World Sequences.** *Slow Flow estimation examples with 100px, 200px, 300px motion magnitudes for the reconstruction dataset.*

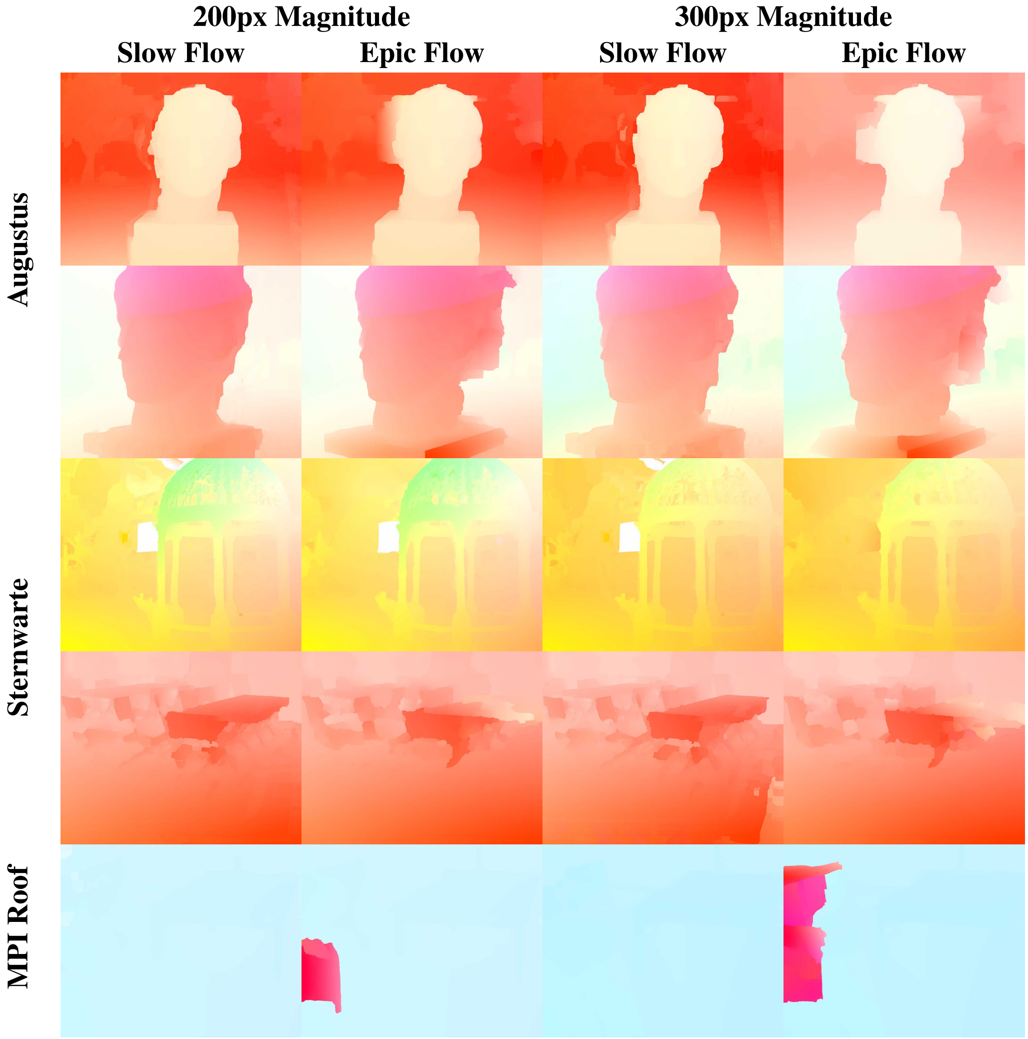


Figure 4.12: **Comparison on Real-World Sequences.** *Comparison of Slow Flow and Epic Flow estimations on the reconstruction dataset.*

that repetitive patterns as the bench moving out of the image in the last row are very troublesome for Epic Flow but can easily be handled by our approach. Furthermore, fine details are better maintained with our approach than Epic Flow, as can be seen with the ceiling of the pavilion in the third and the chairs as well as tables in the fourth row. Our approach is capable of capturing these fine details due to our formulation. The joint flow and occlusion reasoning on HFR sequences based on a variational approach allows for subpixel-accurate flow estimations and sharp motion boundaries. The dense tracking formulation combines these Flowlets while reducing the drift and modeling the occlusions. In contrast, EpicFlow interpolates sparse matches between the first and last frames and uses a variational formulation without occlusion reasoning for refinement.

This difference in performance increases even further if we add motion blur to the input images of the baseline, as described in the following section. We conclude that our technique can be used to benchmark optical flow performance in the presence of large displacements where state-of-the-art methods fail.

4.5 Real-World Benchmark

In this section, we leverage our method to create reference flow fields for challenging real-world video sequences. We have recorded 160 diverse real-world sequences of dynamic scenes using the Fastec TS5Q high-speed camera. For each sequence, we have generated reference flow fields using the approach described in the previous sections. We introduce challenges for a thorough evaluation of optical flow approaches. On the one hand, we vary the magnitude of the motion by using different numbers of Flowlets in our optimization such that the 90% quantile of each sequence reaches a value of 100, 200, or 300 pixels motion. By grouping similar motion magnitudes, we are able to isolate the effect of motion magnitude on each algorithm from other influencing factors. On the other hand, we synthesize motion blur as described in the following section to analyze the performance of modern optical flow algorithms in the presence of motion blur. In all evaluations using our benchmark, we exclude saturated regions that do not carry enough information for our method.

In Fig. 4.13, we show some examples of the generated reference flow fields from our benchmark.

4.5.1 Motion Blur

One interesting and challenging property of real-world videos is motion blur. Motion blur is caused by long shutter times and large motions when different observations are integrated into one pixel. While the reduction of the shutter time minimizes the motion blur, often a higher shutter time is necessary due to adverse lighting conditions. Therefore, motion blur is often be observed in real video sequences. Motion blur is challenging for optical flow methods since high-frequency information is lost. Especially, methods based on sparse matches are affected because it becomes harder to extract good features from the images for matching.

In our real-world HFR sequences, we have almost no motion blur due to the short shutter

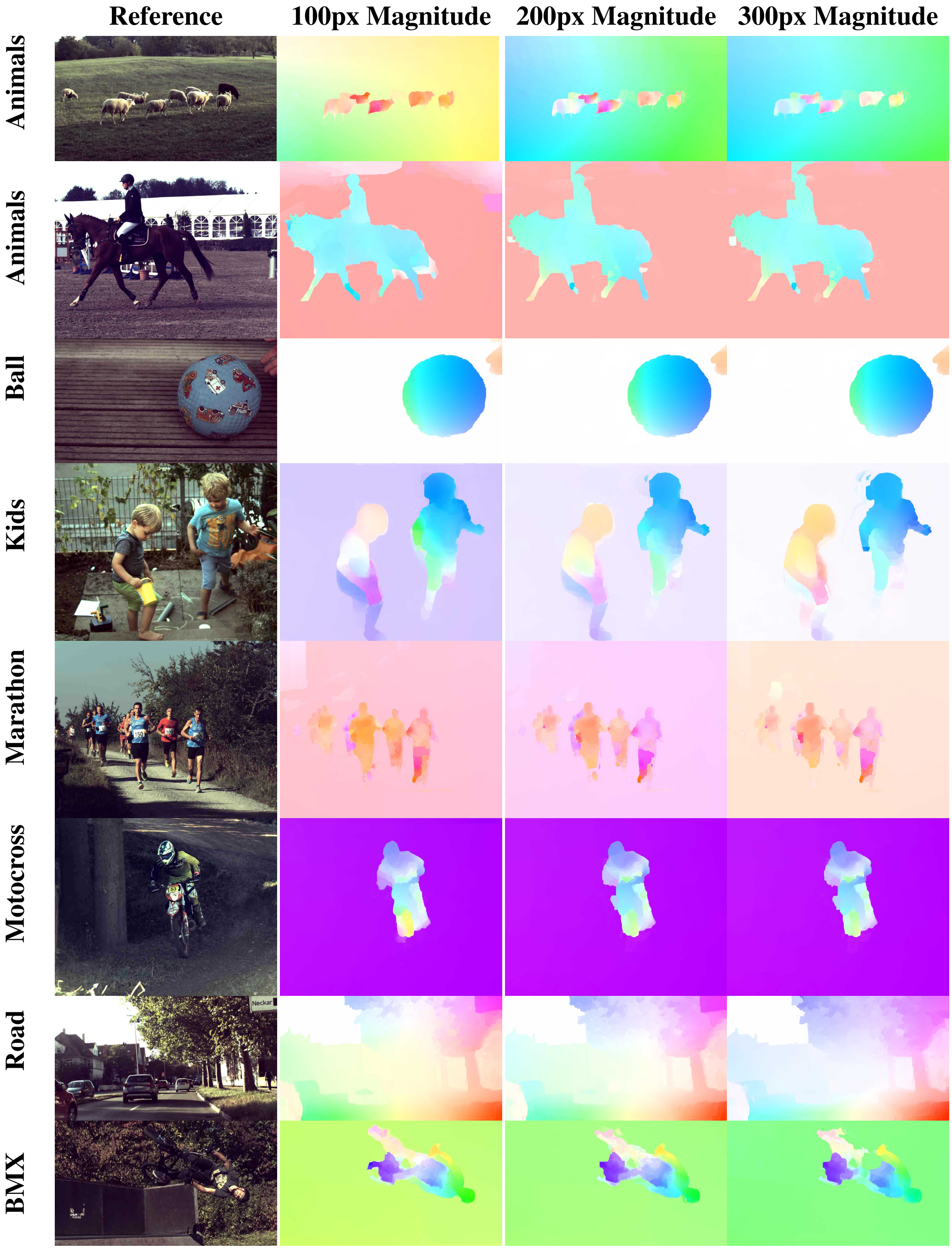
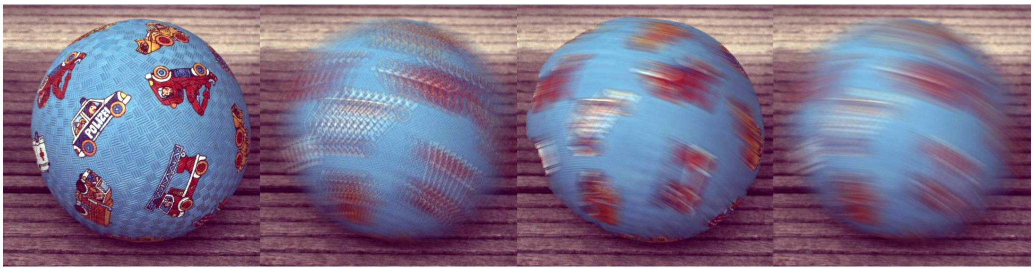


Figure 4.13: **Real-World Reference Data.** *Reference data of different flow magnitudes for some real-world sequences.*



***(a)** Input Image* ***(b)** Simple Average* ***(c)** Low Frame Rate* ***(d)** High Frame Rate*

Figure 4.14: **Motion Blur.** *Using HFR videos and our technique (described in Section 4.2), we are able to add realistic motion blur (d) to the images (a). In contrast, using a simple average over HFR frames (b) or low frame rates with a classical optical flow method results in severe staircase artifacts (c).*

time required for the acquisition. However, we can synthesize the motion blur to make the benchmark more realistic and challenging. Given a HFR sequence, we can approximate the motion blur by averaging over a set of frames. However, the motion in our HFR sequences is still too large (> 1px), which leads to staircase artifacts, as can be seen in Fig. 4.14b. We alleviate this problem by first blurring our frames from the HFR sequence. Given our Flowlets, we can blur each frame according to the pixels motion. In particular, for each reference and target frame of our benchmark, we apply on all neighboring frames in the HFR sequence (in past and future direction) adaptive line-shaped blur kernels obtained from the estimated flow of the corresponding Flowlet. Tracing the corresponding pixels along the optical flow (u,v) for pixel (x,y) of an image $\mathbf{I}$ can be efficiently implemented using Bresenham's line algorithm [Bre65], as described in Algorithm 1. Finally, we approximate the motion blur of each frame of the final frame rate by averaging all blurred neighboring frames. We can control the strength of motion blur by adapting the size of the neighborhood we use in the average.

Changing the neighborhood can be considered as simulating different shutter times since information over shorter or longer periods of time is integrated. As illustrated in Fig. 4.14d, this results in realistic motion blur. For comparison, we also show the blur result when applying the adaptive blur kernel on the low frame rate inputs directly (Fig. 4.14c).

For our benchmark, we consider four different levels of motion blur beside the sharp image, i.e., averaging over 1, 3, 5, or 7 blurred frames. The different level of motion blur for the corresponding reference frames are shown in Fig. 4.15. In scenes Animals, Ball, Kids, and Motocross, the foreground has the dominant motion, which yields a stronger blur on the foreground whereas the scenes BMX and Road also have a quite blurry background. All in all, the motion blur seems very realistic and creates new interesting challenges in our benchmark.

4.5.2 Benchmark

In this section, we benchmark several state-of-the-art techniques on our challenging novel optical flow dataset. We compare 8 state-of-the-art optical flow techniques. More specifically,

Algorithm 1 Blurring based on Bresenham's Line Algorithm

```
function BLURPIXEL($\mathbf{I}, x, y, u, v$)                # pixel (x, y) with flow (u, v)
    $\mathbf{I}_{blurred} = 0$
    $num = 0$                                               # count summed pixels
    $err = abs(u) - abs(v)$                                 # initial error bound
    $i = 0, j = 0$                                          # pixel steps
    while $true$ do
        $\mathbf{I}_{blurred} = \mathbf{I}_{blurred} + \mathbf{I}(x+i, y+j)$      # sum pixels in motion direction
        $num = num + 1$
        if $i > 0$ or $j > 0$ then
            $\mathbf{I}_{blurred} = \mathbf{I}_{blurred} + \mathbf{I}(x-i, y-j)$  # sum pixels in opposite direction
            $num = num + 1$
        end if
        if $i == u$ and $j == v$ then
            return $\mathbf{I}_{blurred}/num$               # return average
        end if
        if $err \geq -0.5 \cdot abs(v)$ then                # large error in x-direction
            $i = i + sign(u)$                               # step in x-direction
            $err = err - abs(v)$                            # add error in y-direction
        end if
        if $err \leq 0.5 \cdot abs(u)$ then                 # large error in y-direction
            $j = j + sign(v)$                               # step in y-direction
            $err = err + abs(u)$                            # add error in x-direction
        end if
    end while
end function
```

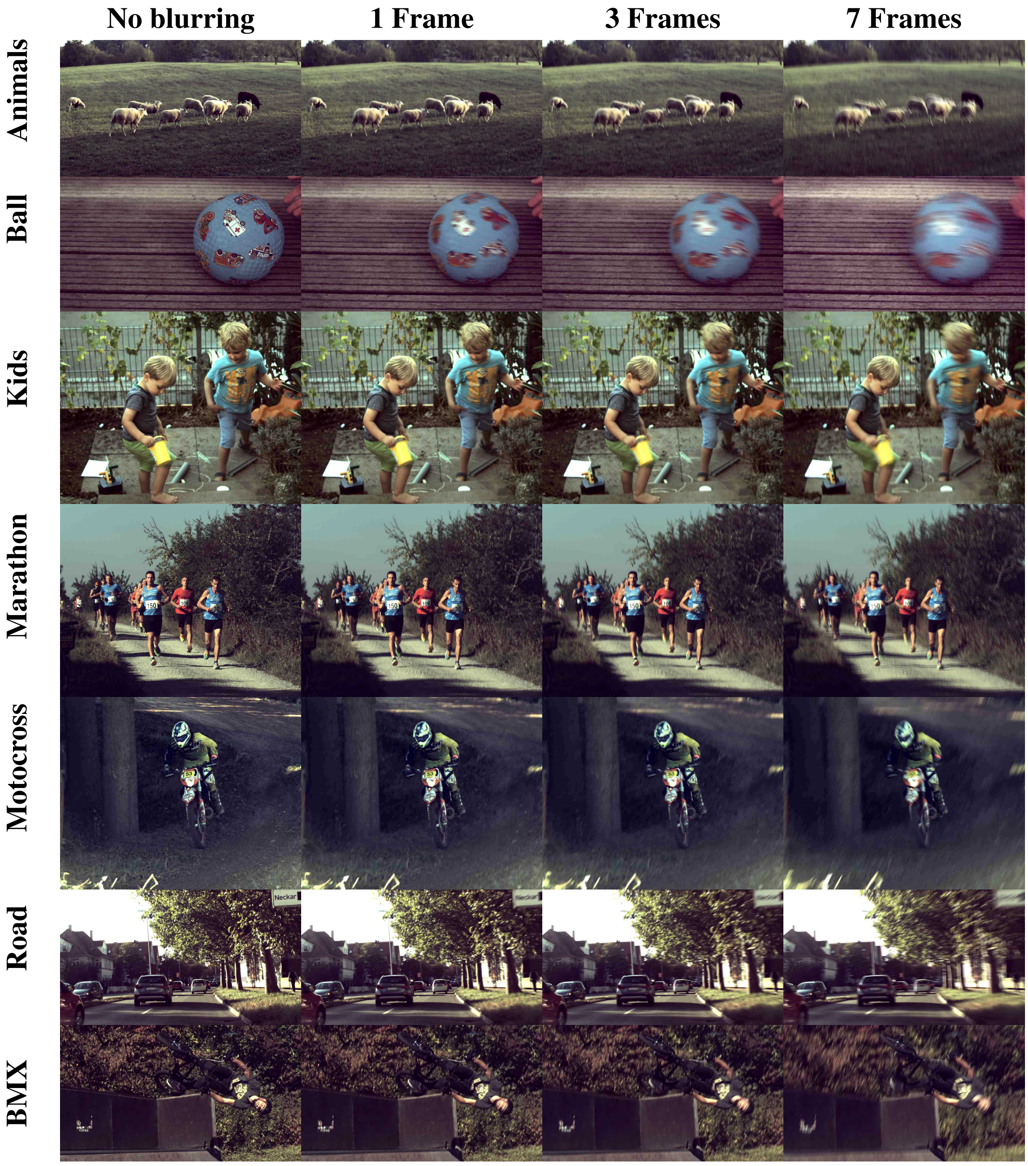


Figure 4.15: **Real-World Motion Blur.** *Different levels of blurring of the reference frame.*

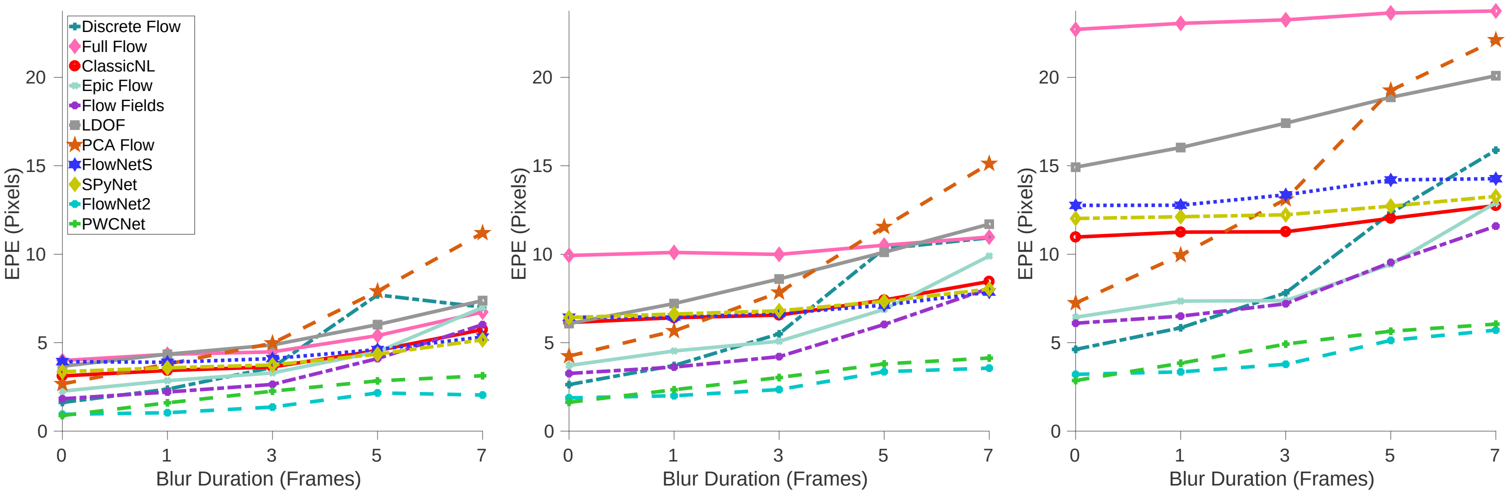
we evaluate DiscreteFlow [MHG15], Full Flow [CK16], ClassicNL [SRB14], EpicFlow [Rev+15], Flow Fields [BTS15], LDOF [BM11], PCA Flow [WB15], FlowNet [Dos+15], SPyNet [RB16], FlowNet2 [Ilg+17], and PWC-Net [Sun+18b; Sun+18a] using the recommended parameter settings, but adapting the maximal displacement to the input. While FlowNet and SPyNet were trained on FlyingChairs [Dos+15], FlowNet2 was trained on FlyingChairs, FlyingThings3D [May+16] and a new version of FlyingChairs with small motions proposed in their paper. Instead, PWC-Net was trained on Sintel [But+12], KITTI [GLU12], HD1K [Kon+16] and Middlebury [Bak+11] as described by Sun et al. [Sun+18a]. We are interested in benchmarking the performance of these methods wrt. two important factors: motion magnitude and motion blur, for which a systematic comparison on challenging real-world data is missing in the literature.

Fig. 4.16 shows our evaluation results in terms of EPE over all sequences. We use three different plots according to the magnitude of the motion ranging from 100 pixels (easy) to 300 pixels (hard). For each plot we vary the length of the blur on the x-axis. The blur length is specified with respect to the number of blurred frames at the highest temporal resolution, where 0 indicates the original sharp images.

As expected, for the simplest case (100 pixels without motion blur), most methods perform well, with FlowNet2 [Ilg+17] and PWC-Net [Sun+18b] outperforming the other baselines. Interestingly, increasing the blur length impacts the methods differently. While matching-based methods like PCA Flow [WB15], EpicFlow [Rev+15] and DiscreteFlow [MHG15] suffer significantly, the performance of learning-based approaches such as FlowNet [Dos+15], SPyNet [RB16], FlowNet2 [Ilg+17], and PWC-Net [Sun+18b] remains largely unaffected. Similarly, only modest loss in performance can be observed for ClassicNL [SRB14], which uses image pyramids instead of feature matches to handle large motions. A similar trend is visible for larger flow magnitudes, where the difference in performance becomes more clearly visible. As expected, the performance of all methods decreases with larger magnitudes. One would expect that approaches based on feature matching are less effected by larger magnitudes than the variational approaches but we can observe a similar drop in performance for all methods. This might be due to scenes with non-rigid objects that contain many cases of self occlusions and affects feature matching approaches as well as variational approaches. We further note that some methods (e.g., Full Flow [CK16]), which perform well on synthetic datasets such as MPI Sintel [But+12] produce large error on our dataset. This underlines the importance of optical flow datasets with real-world images as the one proposed.

In Tables 4.5, 4.6, 4.7, we show the performance on different scene types (columns), and using different levels of motion blur (rows). We grouped our sequences in different scenes with similar objects and motions. In the scenes Motocross, BMX, Rally, Kids and Ball we have only a few objects moving in contrast to Marathon, Town, Road and Animals. Furthermore, the objects in Kids, BMX, Marathon, Animals and Town are mostly non-rigid, which causes complex non-linear motion and self occlusions.

We observe some methods to have particular difficulties in the scenes Motocross, Town, Rally and Road. In Motocross for instance Full Flow [CK16], ClassicNL [SRB14], LDOF [BM11] and FlowNet [Dos+15] achieve around 5 to 6 pixel EPE whereas the others achieve 1 to 3 pixel EPE in the simple case of 100px motion magnitude. This gap still remains



***(a)** 100px Flow Magnitude*

***(b)** 200px Flow Magnitude*

***(c)** 300px Flow Magnitude*

Figure 4.16: **Performance on Real-World Benchmark** *State-of-the-art comparison on the generated reference data wrt. motion magnitude and blur.*

with larger motion magnitudes. In the scene Town the methods Full Flow [CK16], FlowNet [Dos+15] and in Rally the methods Full Flow [CK16], LDOF [BM11] have this kind of difficulties. Besides these difficulties with some scenes, we observe the strongest impact of larger motion magnitudes in scenes with non-rigid objects. For all methods the performance decreases strongly for larger (200 and 300 pixel) motion magnitudes. In BMX, for example, the EPE of the best performing method PWC-Net [Sun+18b] increases from 1.22 to 2.45 and 4.41 pixel. The reasons are complex non-linear motions, appearance changes and self occlusions that become more problematic with larger motion magnitudes. With stronger motion blur, we observe the strongest loss in performance for all methods in the scenes Town and Animals. This is primarily caused by a complex non-linear camera motion that makes it hard to find good matches in the large background region. Only the approaches ClassicNL [SRB14], FlowNet [Dos+15], FlowNet2 [Ilg+17], and PWC-Net [Sun+18b], not using feature matches, are not affected as strongly as the others.

In conclusion, the performance on the different scenes are giving us important insights into the strengths and weaknesses of the different methods. Whereas the methods Full Flow [CK16], ClassicNL [SRB14], LDOF [BM11] and FlowNet [Dos+15] have difficulties with some scenes in general, the motion magnitude has an adversarial effect on all methods. The motion magnitude affects the performance in particular when dealing with non-rigid objects and the motion blur is problematic for feature matching methods, especially with complex camera motion.

4.5.3 Qualitative Results

Finally, we show some qualitative results for DiscreteFlow [MHG15], Epic Flow [Rev+15] and FlowNet2 [Ilg+17] on the 300px magnitude reference data without blur (Fig. 4.17) and with blur length 7 (Fig. 4.18). The illustrations are normalized by the maximum flow of the reference data. The results discussed in the previous section can also be observed in the visualization of the flow fields. Without blur, FlowNet2 is the closest to the reference data in almost all sequences. Increasing the blur length to 7 frames strongly affects the performance of DiscreteFlow and Epic Flow whereas FlowNet2 still achieves good results in all sequences except for BMX.

4.5.4 Remaining Problems

While our analysis in Section 4.4 showed the high accuracy of our method that justified the usage as a benchmark for modern optical flow methods, we see great potential to improve upon our results. In Fig. 4.17, we show some examples of the generated reference flow fields from our benchmark.

In the scenes Ball, Marathon, Motocross, and Road, the resulting flow fields look almost perfect, but small errors are visible in Animals, Kids, and BMX. On the one hand, details are missing, e.g., small parts of the wheel in BMX, and on the other hand, errors occur in occluded regions, e.g., next to the head of the right kid with 300px magnitude and under the head of the horse. Thus, we see great potential in improving our occlusions estimates. We relied on simple linear assumptions in combination with spatial propagation to handle

Blur	Method	Kids	Motocross	BMX	Marathon	Town	Rally	Road	Animals	Ball	Avg
0	Discrete Flow	1.16	2.00	1.71	2.22	0.66	0.63	3.28	0.76	0.21	1.62
	Full Flow	1.22	6.68	1.99	2.61	4.30	6.84	5.33	2.62	2.07	3.99
	ClassicNL	1.67	6.87	3.35	3.82	0.88	1.57	5.51	1.34	1.66	3.12
	Epic Flow	1.57	3.35	2.61	2.84	1.19	1.24	3.97	0.99	0.36	2.26
	Flow Fields	1.19	2.71	1.92	2.69	0.68	0.71	3.63	0.78	0.23	1.83
	LDOF	1.59	5.60	3.27	3.02	1.30	5.85	6.81	1.88	0.72	3.67
	PCA Flow	2.15	2.75	3.17	3.49	1.33	1.70	4.45	1.87	0.66	2.67
	FlowNetS	2.38	5.33	3.75	4.47	3.31	2.44	6.43	1.88	1.91	3.93
	SPyNet	1.70	6.62	3.28	3.83	2.01	1.68	5.80	1.11	2.67	3.36
	FlowNet2	0.77	1.25	1.29	1.22	0.44	0.59	1.51	0.51	0.17	0.95
	PWCNet	0.69	1.76	1.22	1.18	0.51	0.51	1.05	0.44	0.23	0.88
1	Discrete Flow	1.24	2.43	2.02	2.50	3.10	0.99	3.37	1.76	0.32	2.37
	Full Flow	1.29	6.90	2.42	2.93	5.34	7.10	5.32	2.95	1.38	4.34
	ClassicNL	1.73	6.98	3.54	4.03	1.92	1.63	5.49	1.84	1.34	3.43
	Epic Flow	1.63	4.17	2.83	3.14	2.72	1.43	4.40	1.47	0.60	2.85
	Flow Fields	1.28	2.90	2.18	2.91	1.93	0.93	3.54	1.28	0.31	2.21
	LDOF	1.67	6.29	3.71	3.57	4.04	2.31	7.17	3.43	0.76	4.35
	PCA Flow	2.24	3.07	3.50	3.70	5.64	1.85	4.61	3.23	0.70	3.79
	FlowNetS	2.41	5.00	3.82	4.45	3.30	2.22	6.46	1.95	1.51	3.89
	SPyNet	1.77	6.88	3.47	4.05	2.41	1.94	5.77	1.49	3.13	3.58
	FlowNet2	0.78	1.40	1.39	1.34	0.63	0.64	1.53	0.59	0.23	1.04
	PWCNet	0.71	1.86	1.15	1.29	4.09	0.54	1.05	0.80	0.30	1.60
3	Discrete Flow	1.32	3.24	2.30	3.26	7.42	1.22	3.54	3.28	0.54	3.60
	Full Flow	1.40	7.27	2.68	3.02	6.35	6.18	5.13	2.89	1.22	4.48
	ClassicNL	1.79	6.82	3.59	4.22	2.88	1.70	5.38	2.05	1.37	3.63
	Epic Flow	1.66	4.95	3.25	3.29	3.48	1.59	4.58	2.03	2.21	3.29
	Flow Fields	1.32	3.20	2.41	3.32	3.11	1.12	3.71	1.74	0.57	2.64
	LDOF	1.87	6.67	4.01	3.58	5.36	2.28	7.42	4.45	1.04	4.87
	PCA Flow	2.70	4.26	4.06	3.89	9.10	1.96	4.93	4.93	1.15	4.97
	FlowNetS	2.47	5.92	4.03	4.57	3.56	2.23	6.52	2.11	1.62	4.10
	SPyNet	1.85	7.03	3.62	4.11	2.73	2.16	5.77	1.75	3.41	3.74
	FlowNet2	0.80	1.70	1.49	1.47	1.90	0.75	1.55	0.71	0.30	1.36
	PWCNet	0.75	2.00	1.24	1.30	7.31	0.66	1.08	1.00	0.36	2.26
5	Discrete Flow	1.72	4.49	3.55	4.40	23.42	1.79	3.97	7.73	1.00	7.70
	Full Flow	1.73	7.57	3.43	3.88	9.20	4.78	5.50	4.34	1.65	5.40
	ClassicNL	2.12	7.25	4.31	4.84	5.22	2.15	5.50	3.67	1.52	4.56
	Epic Flow	2.03	5.84	3.91	3.99	6.63	1.98	4.86	3.75	3.68	4.48
	Flow Fields	1.68	4.75	3.59	4.07	6.54	1.65	4.29	3.70	1.20	4.10
	LDOF	2.24	7.73	5.06	4.10	7.94	2.57	7.80	6.31	1.63	6.02
	PCA Flow	3.34	5.32	5.44	5.20	18.08	2.51	5.20	10.38	1.72	7.91
	FlowNetS	2.64	6.36	4.31	4.99	5.07	2.49	6.54	2.67	1.82	4.61
	SPyNet	2.10	7.61	4.10	4.50	4.07	2.68	5.98	2.53	3.57	4.35
	FlowNet2	0.91	1.99	1.64	1.76	5.42	0.85	1.65	1.06	0.40	2.16
	PWCNet	0.90	2.49	1.50	1.59	8.99	0.84	1.15	1.91	0.53	2.84
7	Discrete Flow	2.33	5.99	5.12	4.92	14.71	2.42	4.60	9.87	1.58	7.01
	Full Flow	2.17	8.37	4.61	4.77	11.95	5.46	5.88	6.86	2.07	6.73
	ClassicNL	2.56	8.13	5.28	5.55	7.90	2.85	5.73	5.69	2.00	5.74
	Epic Flow	2.58	7.16	4.98	5.03	15.38	2.62	5.21	6.43	4.95	6.99
	Flow Fields	2.29	6.03	4.82	6.04	10.91	2.51	4.69	7.16	1.82	6.02
	LDOF	2.83	8.79	6.12	4.94	11.03	3.20	8.22	8.36	2.56	7.38
	PCA Flow	4.54	7.99	7.84	8.24	23.86	3.36	5.75	18.57	2.90	11.20
	FlowNetS	2.92	7.17	4.66	5.56	6.89	3.03	6.67	3.55	2.15	5.32
	SPyNet	2.48	8.33	4.70	5.05	5.99	3.24	6.26	3.50	3.76	5.16
	FlowNet2	1.09	2.42	1.98	2.03	3.45	1.06	1.79	1.59	0.51	2.04
	PWCNet	1.14	3.33	2.13	2.09	8.11	1.16	1.27	3.23	0.91	3.13

Table 4.5: **Performance on Real-World Scenes** *State-of-the-art comparison on the generated reference data with 100 pixel motion magnitude wrt. motion blur.*

Blur	Method	Kids	Motocross	BMX	Marathon	Town	Rally	Road	Animals	Ball	Avg
0	Discrete Flow	2.44	5.26	3.79	3.96	0.87	0.59	3.58	1.51	0.41	2.63
	Full Flow	4.56	15.12	8.99	8.42	4.69	14.94	9.95	15.15	10.40	9.93
	ClassicNL	4.12	12.18	8.27	8.99	1.44	2.55	8.57	3.55	9.33	6.14
	Epic Flow	3.30	6.55	5.42	5.14	1.52	0.96	5.41	1.78	1.49	3.71
	Flow Fields	2.62	4.77	4.39	5.90	0.93	0.61	5.39	1.67	0.89	3.26
	LDOF	3.67	9.10	6.57	5.60	1.78	6.41	10.96	2.90	8.48	6.08
	PCA Flow	3.78	6.18	6.36	5.28	1.66	1.68	6.18	2.79	1.97	4.24
	FlowNetS	4.84	10.77	7.21	8.67	3.88	3.48	8.84	3.85	6.09	6.44
	SPyNet	3.90	14.40	7.16	7.79	2.61	2.35	9.17	3.41	11.43	6.42
	FlowNet2	1.43	2.52	2.52	2.44	0.59	1.13	3.33	0.86	0.60	1.88
	PWCNet	1.27	3.72	2.45	2.31	0.63	0.44	2.08	0.74	0.50	1.63
1	Discrete Flow	2.50	7.61	4.21	4.15	3.20	1.02	3.78	3.51	1.01	3.71
	Full Flow	4.59	13.94	8.90	8.82	5.73	14.52	10.06	15.49	11.27	10.10
	ClassicNL	4.13	11.84	8.60	9.37	2.46	2.66	8.44	4.06	8.36	6.41
	Epic Flow	3.47	7.93	5.84	7.04	3.10	1.17	5.45	2.34	5.09	4.53
	Flow Fields	2.52	5.36	4.26	5.53	2.08	0.95	5.45	2.37	1.43	3.62
	LDOF	3.78	10.56	7.35	6.42	4.61	2.85	11.12	6.90	7.83	7.21
	PCA Flow	4.16	6.57	6.61	6.80	6.03	1.78	6.73	4.62	2.90	5.66
	FlowNetS	4.84	10.81	7.37	8.82	3.86	3.40	8.73	4.04	5.39	6.45
	SPyNet	3.92	14.57	7.47	7.99	2.98	2.60	9.09	3.72	11.70	6.62
	FlowNet2	1.42	2.91	2.83	2.61	0.76	1.39	3.20	0.98	0.47	2.00
	PWCNet	1.28	3.77	2.27	2.56	4.07	0.54	2.11	1.15	0.65	2.34
3	Discrete Flow	2.97	7.35	4.93	6.11	7.34	1.34	4.20	8.03	2.29	5.50
	Full Flow	4.62	14.20	8.76	8.65	6.71	12.40	10.00	14.84	10.33	9.99
	ClassicNL	4.23	11.80	8.59	9.51	3.43	2.74	8.18	4.23	7.94	6.56
	Epic Flow	3.65	8.62	6.10	6.58	3.93	1.42	5.85	3.36	9.04	5.08
	Flow Fields	2.87	6.28	4.91	7.03	3.40	1.05	5.21	2.78	2.50	4.21
	LDOF	4.29	12.93	8.87	6.61	6.09	2.89	11.79	10.53	7.90	8.60
	PCA Flow	5.12	7.85	10.61	7.67	11.36	1.91	6.69	7.26	4.02	7.84
	FlowNetS	4.91	11.20	7.67	9.07	4.12	3.50	8.63	4.50	5.46	6.65
	SPyNet	3.97	14.79	7.67	8.18	3.29	2.79	9.07	3.94	12.25	6.80
	FlowNet2	1.45	3.52	2.73	2.72	2.05	1.42	3.22	1.29	0.66	2.35
	PWCNet	1.32	4.07	2.28	2.59	6.68	0.68	2.22	2.18	0.76	3.03
5	Discrete Flow	3.68	11.00	8.33	7.93	19.97	1.92	4.96	17.02	4.12	10.31
	Full Flow	4.64	15.19	9.92	8.77	9.62	9.67	9.80	14.76	9.43	10.51
	ClassicNL	4.42	12.59	9.14	9.95	5.70	3.40	8.17	5.57	7.78	7.41
	Epic Flow	4.18	12.39	6.69	7.11	7.69	1.68	6.31	6.40	12.24	6.88
	Flow Fields	3.08	8.52	6.76	7.58	6.45	1.66	5.74	6.70	4.33	6.03
	LDOF	5.00	14.92	11.63	7.30	8.71	3.35	11.81	12.92	7.72	10.11
	PCA Flow	6.59	9.04	11.92	9.85	19.58	3.41	7.57	16.50	6.71	11.55
	FlowNetS	5.18	11.49	8.05	9.65	5.59	3.68	8.59	4.80	5.74	7.12
	SPyNet	4.14	14.81	8.19	8.74	4.65	3.24	9.10	4.64	13.24	7.34
	FlowNet2	1.53	4.32	2.83	3.08	5.53	1.06	3.35	3.05	0.93	3.36
	PWCNet	1.45	4.65	2.77	2.86	8.51	0.76	2.29	3.92	1.46	3.80
7	Discrete Flow	4.50	10.82	11.06	9.12	16.49	2.81	5.63	20.52	6.55	10.93
	Full Flow	5.01	15.43	10.00	8.73	12.18	8.62	10.24	14.19	9.68	10.97
	ClassicNL	4.78	13.55	10.06	10.55	8.21	4.47	8.14	7.19	8.17	8.46
	Epic Flow	4.74	14.66	7.80	7.62	14.28	2.46	6.76	15.12	13.01	9.90
	Flow Fields	3.86	10.45	7.50	8.96	11.34	3.04	6.08	10.25	6.13	8.07
	LDOF	5.63	16.44	13.22	8.39	11.61	4.61	12.32	15.49	8.63	11.70
	PCA Flow	7.69	12.93	15.59	12.65	28.13	4.03	8.31	20.76	9.18	15.12
	FlowNetS	5.59	12.02	8.52	10.46	7.40	4.06	8.74	5.67	6.54	7.86
	SPyNet	4.41	15.02	8.68	9.51	6.54	3.78	9.10	5.52	14.05	8.03
	FlowNet2	1.68	5.18	3.11	3.43	3.54	1.31	3.42	5.82	1.15	3.56
	PWCNet	1.64	4.90	3.50	3.26	8.68	1.15	2.32	4.52	2.34	4.13

Table 4.6: **Performance on Real-World Scenes** *State-of-the-art comparison on the generated reference data with 200 pixel motion magnitude wrt. motion blur.*

Blur	Method	Kids	Motocross	BMX	Marathon	Town	Rally	Road	Animals	Ball	Avg
0	Discrete Flow	4.74	7.27	6.31	5.69	1.34	1.38	7.43	2.65	4.72	4.62
	Full Flow	9.43	29.38	21.68	14.34	14.28	25.11	22.05	43.26	11.68	22.71
	ClassicNL	7.60	14.84	13.55	16.44	4.99	3.27	16.73	8.19	10.59	10.97
	Epic Flow	6.11	8.02	8.80	7.45	2.23	1.62	11.38	3.85	5.48	6.44
	Flow Fields	4.84	7.63	7.59	9.47	1.46	0.89	12.07	2.66	5.06	6.10
	LDOF	7.44	14.91	12.92	13.28	4.41	6.84	21.15	31.44	10.50	14.92
	PCA Flow	6.93	9.11	9.18	7.47	2.47	2.20	13.40	4.38	6.00	7.24
	FlowNetS	8.56	17.11	12.83	15.07	6.34	5.31	16.29	18.86	8.75	12.76
	SPyNet	7.71	17.95	12.38	14.43	5.95	3.48	17.36	13.55	12.21	12.02
	FlowNet2	2.44	4.09	3.73	3.60	0.87	1.38	6.92	1.35	0.60	3.21
	PWCNet	2.47	4.85	4.41	3.90	0.85	0.77	4.20	1.89	0.50	2.86
1	Discrete Flow	5.07	7.59	7.59	6.67	3.67	1.96	7.75	5.30	5.05	5.84
	Full Flow	9.75	28.73	21.80	14.37	15.22	25.82	22.23	43.83	11.97	23.05
	ClassicNL	7.58	14.81	13.89	16.62	6.06	3.50	16.78	8.33	9.94	11.25
	Epic Flow	6.37	9.60	8.96	9.12	3.92	1.84	12.18	4.25	7.61	7.35
	Flow Fields	4.97	7.33	7.73	10.69	2.86	1.39	11.54	3.45	5.34	6.50
	LDOF	7.68	15.97	14.28	13.94	8.99	3.60	21.37	32.49	9.93	16.03
	PCA Flow	7.79	9.25	9.74	10.42	8.85	3.07	13.63	11.85	6.63	9.95
	FlowNetS	8.70	17.40	12.68	15.21	6.00	4.95	16.08	19.79	8.16	12.77
	SPyNet	7.72	18.37	12.60	14.49	6.21	3.76	17.18	13.43	12.62	12.12
	FlowNet2	2.48	4.34	3.87	3.71	1.08	1.50	6.74	1.59	2.56	3.35
	PWCNet	2.51	4.73	3.84	3.88	5.52	0.84	4.18	3.10	2.66	3.84
3	Discrete Flow	5.59	8.71	8.18	9.70	10.40	2.24	7.77	7.00	6.09	7.82
	Full Flow	10.03	29.65	21.44	14.78	15.84	24.35	22.42	44.49	11.37	23.25
	ClassicNL	7.56	14.71	13.93	15.98	7.30	3.60	16.57	7.49	9.55	11.27
	Epic Flow	6.95	10.32	8.55	8.26	4.30	2.06	11.59	4.57	9.59	7.38
	Flow Fields	5.70	9.69	8.18	10.99	4.20	1.50	11.61	4.00	6.19	7.20
	LDOF	8.42	18.86	15.70	14.49	11.54	3.95	21.93	34.19	9.92	17.41
	PCA Flow	10.12	10.24	11.48	11.75	18.95	3.18	13.85	16.51	7.51	13.11
	FlowNetS	9.02	17.96	13.10	15.94	7.13	5.14	15.95	21.27	8.25	13.36
	SPyNet	7.83	18.75	12.87	14.60	6.55	4.03	17.05	13.13	13.01	12.23
	FlowNet2	2.57	5.24	4.16	4.27	1.83	1.84	6.62	2.39	2.68	3.78
	PWCNet	2.65	4.65	3.96	3.99	9.98	1.01	4.28	4.48	2.74	4.92
5	Discrete Flow	7.18	11.18	11.34	11.86	18.39	3.40	9.14	19.43	7.18	12.33
	Full Flow	10.35	30.34	22.48	15.60	17.93	18.85	22.30	45.90	10.90	23.64
	ClassicNL	7.79	15.40	14.75	16.37	9.64	4.47	16.25	8.32	9.45	12.03
	Epic Flow	7.89	14.27	10.01	9.82	8.03	2.64	12.25	7.27	13.03	9.43
	Flow Fields	6.33	13.63	9.82	13.15	7.77	3.79	12.53	7.66	7.45	9.54
	LDOF	9.83	19.75	19.61	15.24	14.22	5.32	22.13	34.62	9.65	18.87
	PCA Flow	10.91	12.53	18.00	22.38	31.03	4.72	15.06	27.50	9.24	19.26
	FlowNetS	9.56	18.09	13.84	16.32	10.86	5.30	15.79	21.24	8.41	14.20
	SPyNet	8.09	19.35	13.35	15.03	7.87	4.67	17.01	13.26	13.73	12.72
	FlowNet2	2.78	5.85	4.89	4.54	5.12	1.62	6.69	6.47	2.85	5.13
	PWCNet	2.90	5.48	4.86	4.69	11.14	1.30	4.37	5.93	3.14	5.65
7	Discrete Flow	8.30	12.48	12.69	11.99	21.10	4.83	10.71	35.31	9.02	15.88
	Full Flow	10.83	29.48	22.88	16.45	19.06	16.31	22.64	45.52	11.36	23.75
	ClassicNL	8.20	16.22	15.44	17.20	11.70	5.47	15.87	8.87	9.81	12.75
	Epic Flow	8.32	16.49	10.75	11.53	19.46	3.49	12.83	12.49	13.86	12.91
	Flow Fields	8.46	12.93	11.94	14.15	13.71	4.47	12.72	10.45	8.81	11.59
	LDOF	10.54	21.29	22.36	15.33	16.18	8.13	22.26	34.78	10.28	20.09
	PCA Flow	12.41	13.80	23.17	16.25	33.66	5.23	16.33	39.06	11.17	22.11
	FlowNetS	10.18	18.59	14.31	16.78	12.69	5.60	15.77	17.72	9.00	14.27
	SPyNet	8.40	19.98	13.91	15.56	9.41	5.32	16.96	13.42	14.31	13.27
	FlowNet2	3.02	6.44	5.87	4.94	6.05	1.86	6.88	7.23	3.07	5.71
	PWCNet	3.30	6.05	5.68	5.16	10.82	2.02	4.73	6.51	3.75	6.05

Table 4.7: **Performance on Real-World Scenes** *State-of-the-art comparison on the generated reference data with 300 pixel motion magnitude wrt. motion blur.*

occlusions. However, more sophisticated motion models will improve the results. Especially, learning-based methods have the advantage that such motion models can be learned from data directly. The limited annotations can be resolved by relying on semi-supervised or unsupervised schemes, as discussed in the next chapter.

In addition, the consideration of a confidence measure for the generated reference data seems very promising. Such a measure allows reducing or even excluding less reliable estimates from the evaluation. Ideally, a probabilistic version of our approach could simultaneously provide point estimates and a measure of confidence.

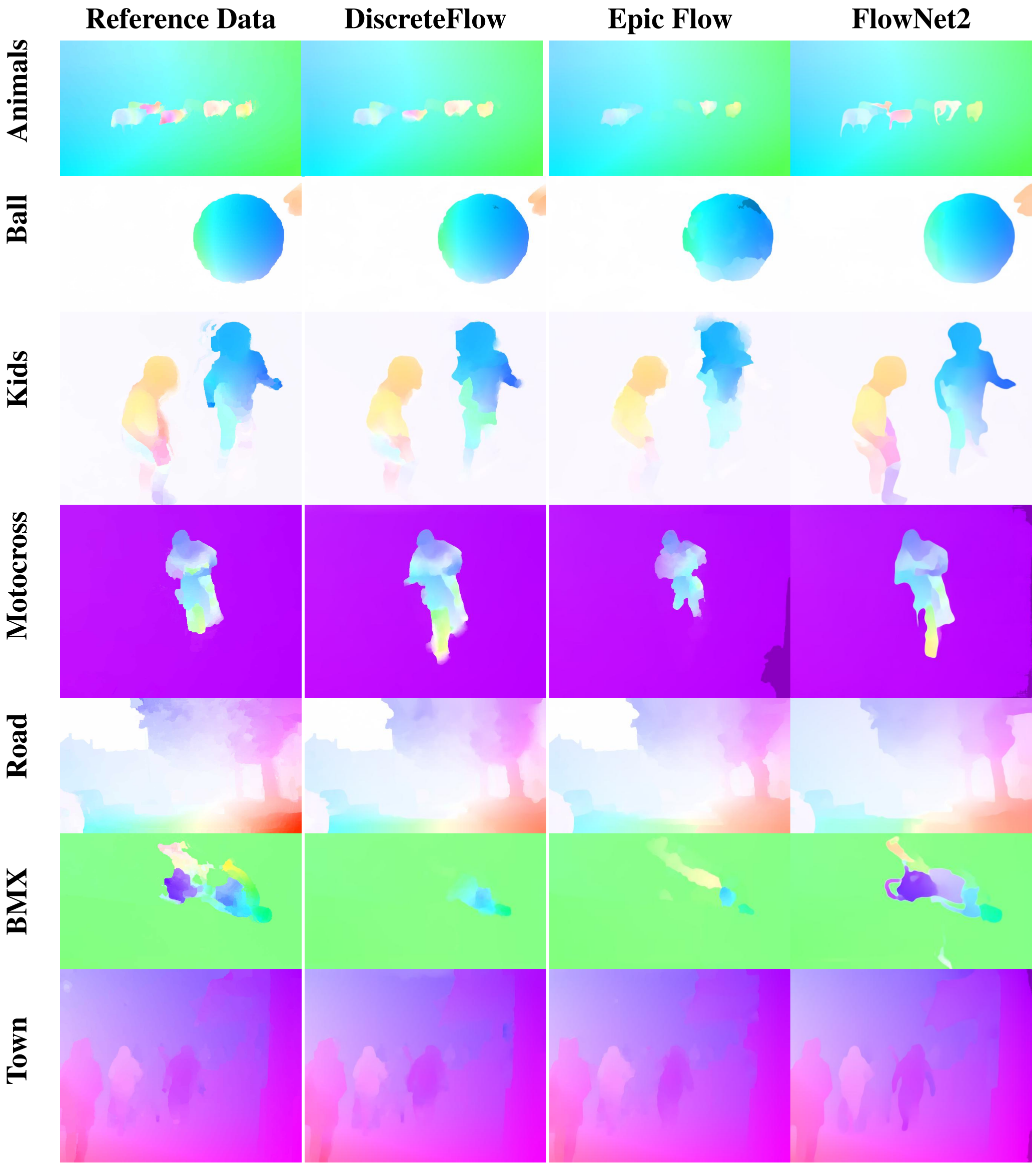


Figure 4.17: *Comparison of DiscreteFlow, Epic Flow and FlowNet2 without blur to the reference data of 300px magnitude.*

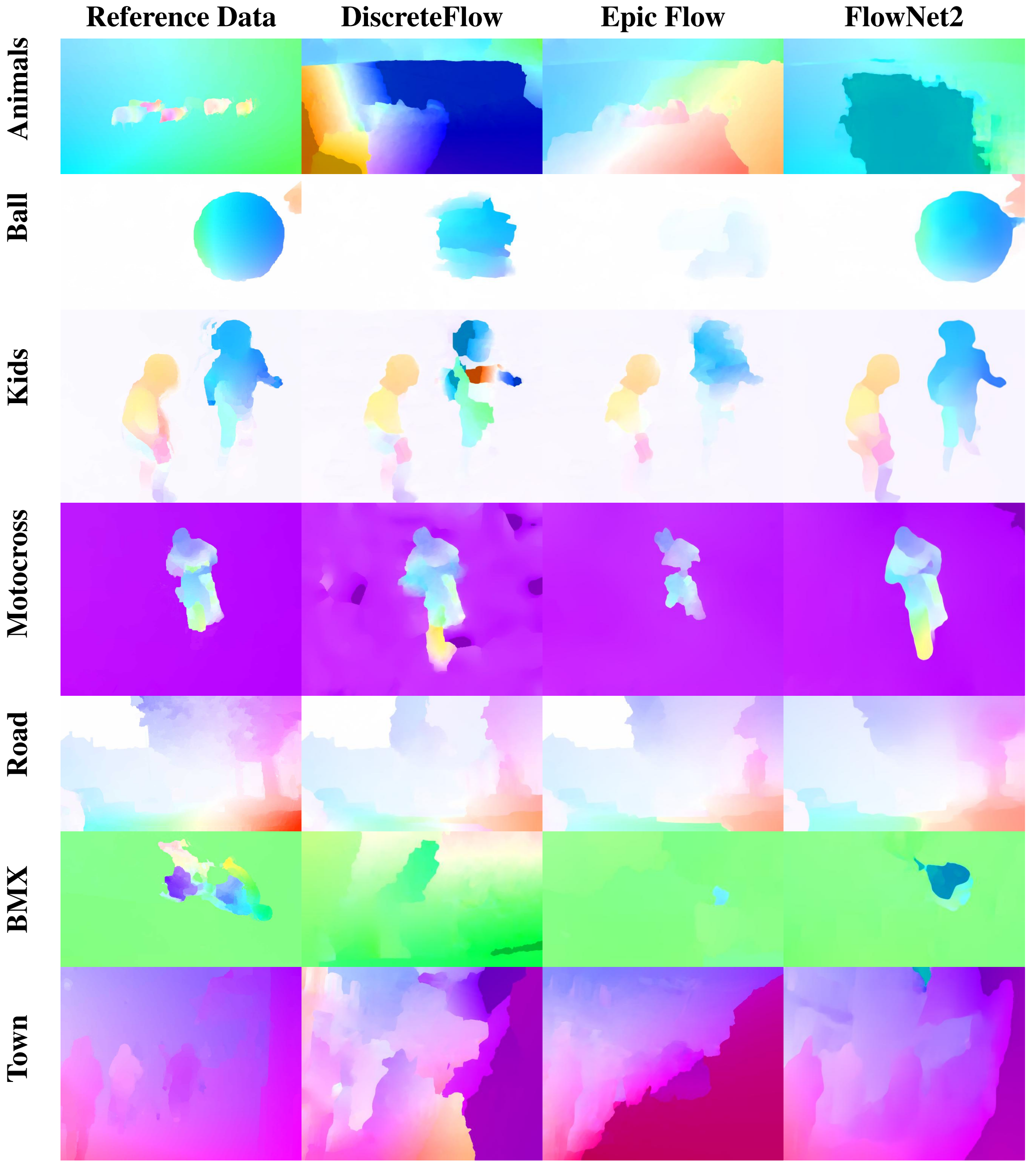


Figure 4.18: *Comparison of DiscreteFlow, Epic Flow and FlowNet2 with 7 frames blur length to the reference data of 300px magnitude.*

5 Unsupervised Learning of Multi-Frame Flow and Occlusions

Learning to solve optical flow in an end-to-end fashion [Dos+15; RB16; Ilg+17; Sun+18b] from examples is attractive as deep neural networks allow for learning more complex hierarchical flow representations directly from annotated data. However, training such models requires large datasets, and obtaining ground truth for real images is challenging. While the approach described in Chapter 4 allows creating new real-world datasets, the process is still time-consuming and might incorporate errors. In addition, deep models trained in a supervised fashion suffer from the overfitting problem. The best performance of modern methods is usually achieved by fine-tuning on the training set of the dataset used for evaluation, and the fine-tuned models perform very poorly on other datasets. Thus, it is essential to train modern optical flow methods on data that follows a similar distribution as the data where the method will be applied eventually.

Alternatively, learning optical flow can be framed as an unsupervised learning problem. In contrast to supervised learning, any video sequence can be used for learning without relying on optical flow annotations. Thus, large video collections from the web can be leveraged to learn strong models. In addition, unsupervised learning enables fine-tuning models on data of the final application since annotations are not necessary.

Learning optical flow in an unsupervised fashion is usually realized by minimizing a photometric loss [YHD16; Vij+17; PHC16; Ren+17; Wan+18b; MHR18], measuring how well the predicted flow warps the target image to the reference frame. Particularly problematic in this setting are occluded regions [Wan+18b; MHR18], which provide misleading information to the photometric loss function. This problem is illustrated in Fig. 5.1, with an example from the synthetic MPI Sintel dataset [But+12]. The photometric loss compares the reference image (Fig. 5.1(b)) to the target image that is warped according to the optical flow estimate (Fig. 5.1(d)). Note that occluded regions in the target image cannot be recovered correctly, even when using the ground truth optical flow field (Fig. 5.1(e)). Instead, the so-called *ghosting effects* occur, i.e., parts of the occluder remain visible in the occluded regions. Recent works [Wan+18b; MHR18] propose to exclude these regions in the photometric loss by inferring occluded regions using the backward flow, i.e., the flow from the target frame to the reference frame. However, these approaches depend heavily on an accurate flow prediction and use heuristics (e.g., thresholding) to infer occlusions.

We propose to model temporal relationships over multiple frames in order to learn optical flow and occlusions jointly. For this purpose, we extend the two-frame architecture proposed by Sun et al. [Sun+18b] to multiple frames. We estimate optical flow in both past and future direction together with an occlusion map within a temporal window of three frames. The occlusion map encodes the state of each pixel as either visible in the past, future, or

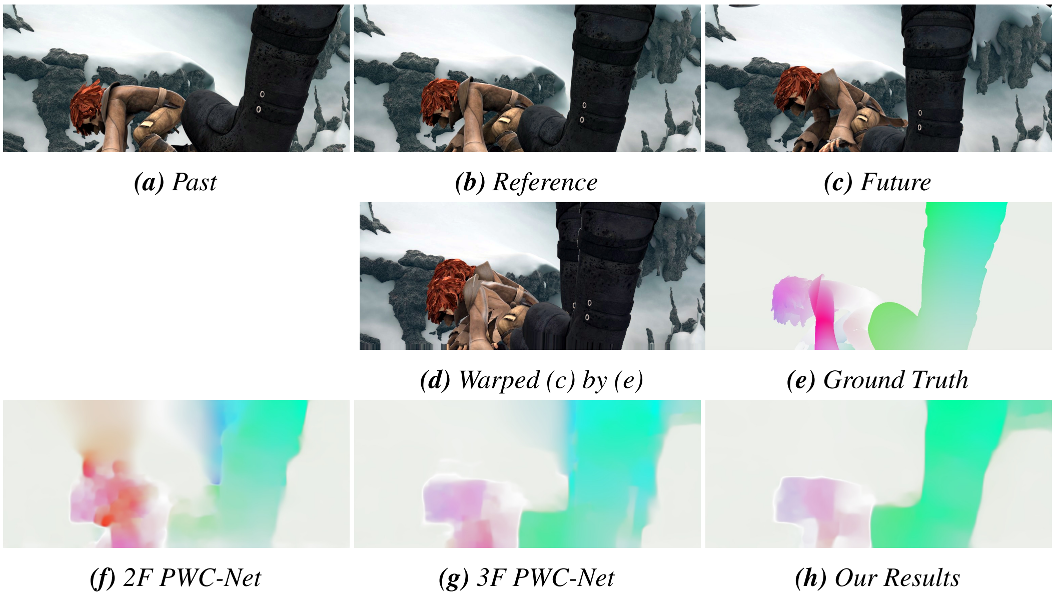

(a) *Past* ***(b)*** *Reference* ***(c)*** *Future*
(d) *Warped (c) by (e)* ***(e)*** *Ground Truth*
(f) *2F PWC-Net* ***(g)*** *3F PWC-Net* ***(h)*** *Our Results*

Figure 5.1: **Motivation.** *Unsupervised optical flow estimation is challenging as commonly used photometric terms are violated in occluded regions. This example from our MPI Sintel [But+12] illustrates the problem of ghosting effects (d) when warping the target frame (c) according to the true flow (e). Classical two-frame approaches produce blurry results near occlusion boundaries (f). Using multiple frames without occlusion reasoning neither solves the problem (g). In contrast, our multi-frame model with explicit occlusion reasoning leads to accurate flow estimates with sharp boundaries (h).*

throughout the temporal window. Our unsupervised loss evaluates the warped images from the past and the future based on the estimated flow fields and occlusion map. In addition to typical spatial smoothness constraints, we introduce a constant velocity constraint within the temporal window. This allows reasoning about occlusions in a principled manner while leveraging temporal information for more accurate optical flow prediction in occluded regions.

In ablation studies (Section 5.4.3) performed on our RoamingImages dataset, which we have created based on randomly moving image patches from Flickr, we show the advantage of our formulation in comparison to a two-frame and multi-frame formulation without occlusion modeling. Eventually in Section 5.4.5, we compare our approach on the popular datasets KITTI 2015 [Gei+13; MG15] and MPI Sintel [But+12] to the state of the art in unsupervised as well as supervised learning of optical flow. Surprisingly, our model trained only on the simplistic RoamingImages dataset outperforms all previous unsupervised optical flow approaches trained on FlyingChairs. By unsupervised fine-tuning on the respective training sets, we further improve our results, reducing the gap to several supervised methods.

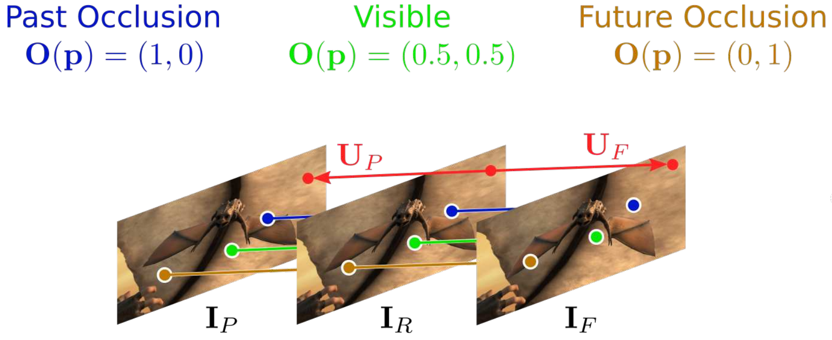


Figure 5.2: **Unsupervised Formulation.** *Illustration of our multi-frame formulation with the past* $\mathbf{U}_P(\mathbf{p})$*, future flow* $\mathbf{U}_F(\mathbf{p})$ *and occlusions* $\mathbf{O}(\mathbf{p})$*. The green pixel is visible in all frames, the blue pixel is occluded in the past* $\mathbf{I}_P$*, and the brown pixel is occluded in the future* $\mathbf{I}_F$*.*

5.1 Formulation

Now, we develop an approach for unsupervised learning of optical flow and occlusions by leveraging multiple frames. In unsupervised learning of optical flow, only the photometric loss provides guidance. The photometric loss warps the target frame according to the flow estimate and compares the warped target frame to the reference frame. Local ambiguities caused by untextured regions are handled with an additional spatial smoothness constraint that propagates information between neighboring pixels. However, learning optical flow in an unsupervised fashion is complicated due to ambiguities caused by non-Lambertian reflectance, occlusions, large motions, and illumination changes. Considering multiple frames can help to resolve some of the ambiguities, in particular, those caused by occlusions. We thus develop a multi-frame formulation to train a convolutional neural network to predict flow fields and occlusions jointly.

Let $\mathcal{I} = \{\mathbf{I}_P, \mathbf{I}_R, \mathbf{I}_F\}$ denote three consecutive RGB frames $\mathbf{I}_t \in \mathbb{R}^{W \times H \times 3}$ from video with a standard frame rate. Our goal is to predict the optical flow $\mathbf{U}_F \in \mathbb{R}^{W \times H \times 2}$ from reference frame $\mathbf{I}_R$ to future frame $\mathbf{I}_F$ while leveraging past frame $\mathbf{I}_P$. In this short temporal window, we assume the motion to be approximately linear. The simplest way to enforce a linear motion is using a hard constraint, as in Section 4.2, by predicting only one flow field and warping both images $\mathbf{I}_P, \mathbf{I}_F$ to reference image $\mathbf{I}_R$ according to this flow field for computing the photometric loss. However, realistic scenes usually contain more complex motions, which violate this hard constraint (e.g., road surface in KITTI). Therefore, we formulate a soft constraint by predicting two optical flow fields and encouraging constant velocity: We denote $\mathbf{U}_F$ the flow field from reference frame $\mathbf{I}_R$ to future frame $\mathbf{I}_F$, and $\mathbf{U}_P \in \mathbb{R}^{W \times H \times 2}$ the flow field from reference frame $\mathbf{I}_R$ to past frame $\mathbf{I}_P$.

Regardless of the motion model, photo-consistency is violated in occluded regions. Considering three frames allows us to resolve this problem by reasoning about occlusions in a data-driven fashion, as illustrated in Fig. 5.2. Let us consider a pixel $\mathbf{p}$ in reference frame $\mathbf{I}_R$. Note that, by definition, the pixel is visible in the reference frame. Thus, there

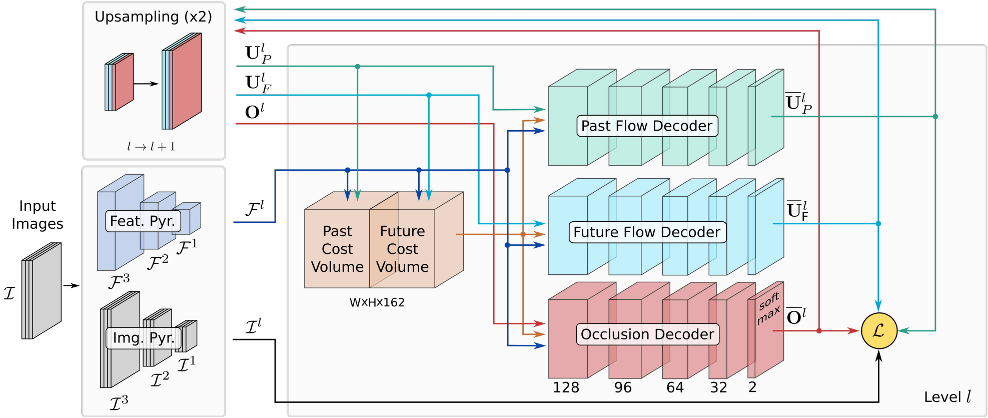


Figure 5.3: **Network Architecture.** *Given the input sequence $\mathcal{I}$, we construct an image and a feature pyramid. The optical flow is estimated in a coarse-to-fine manner: at level l, two cost volumes are constructed from the features $\mathcal{F}^l$ of the past and future frame, warped according to the current optical flow estimates $\mathbf{U}^l_P$ and $\mathbf{U}^l_F$, respectively. The two cost volumes are decoded resulting in the past flow $\overline{\mathbf{U}}^l_P$, future flow $\overline{\mathbf{U}}^l_F$, and an occlusion map $\overline{\mathbf{O}}^l$ at level l. The estimations are passed to the upsampling block to yield inputs for the next level $l+1$ of the pyramid. See text for details.*

are only three possible cases: Either it is visible in all frames, or it has been occluded in the past, or it becomes occluded in the future. While there exists a possible fourth state, i.e., when a pixel is solely visible in the reference frame, this is a very unusual case that rarely occurs in practice and, therefore, can be discarded. Thus, the occlusion of each pixel can be represented with three states. The advantage of the multi-frame formulation is that we have observations in all three cases, which we can use to reason about the optical flow and occlusions. More formally, we model occlusions by introducing a continuous occlusion variable $\mathbf{O} \in [0,1]^{W \times H \times 2}$ at every pixel, which allows us to correctly evaluate the photometric loss by reducing the importance of occluded pixels. Let $\mathbf{O}(\mathbf{p}) \in [0,1]^2$ denote the occlusion at pixel $\mathbf{p}$ where $\|\mathbf{O}(\mathbf{p})\|_1 = 1$. If $\mathbf{O}(\mathbf{p}) = (1,0)$, we consider $\mathbf{p}$ as occluded in the past, if $\mathbf{O}(\mathbf{p}) = (0,1)$, pixel $\mathbf{p}$ is occluded in the future and if $\mathbf{O}(\mathbf{p}) = (0.5,0.5)$, pixel $\mathbf{p}$ is visible in all frames. While continuous occlusion variables make the formulation differentiable, it also allows modeling pixels that are only partially occluded.

We propose to estimate $\mathbf{U}_F$, $\mathbf{U}_P$, and $\mathbf{O}$ jointly using a neural network while enforcing $\|\mathbf{O}(\mathbf{p})\|_1 = 1$ with a softmax at the last layer of the network.

5.2 Network Architecture

As discussed in Section 3.4, several network architectures have been proposed for the optical flow problem. The recently proposed PWC-Net architecture [Sun+18b] borrows ideas from the classical optical flow and stereo methods. They extract meaningful features $\mathcal{F}^l = \{\mathbf{F}^l_R, \mathbf{F}^l_F\}$ using a siamese network that consists of two convolutional neural networks with shared weights, one processing the reference frame ($\mathbf{F}_R^l$) and the other processing the future frame ($\mathbf{F}_F^l$). Using strided convolutions, they extract features on different resolutions while increase the number of output channels (16, 32, 64, 128, 192). In a coarse-to-fine approach, they address a residual estimation problem on each scale by upsampling the optical flow of the lower scale and warping the future features accordingly. In a so-called cost volume, they correlate the features of the reference frame and the warped features of the future frame. In specific, for each location $\mathbf{p}$ in the reference frame, they compute the inner product between the feature of the reference frame and a local neighborhood $\mathcal{N}$ around $\mathbf{p}$ of the warped future feature map.

$$\mathcal{C}(\mathbf{F}_R^l, \mathbf{F}_F^l, \mathbf{p}) = \{\frac{1}{|\mathcal{N}|} \langle \mathbf{F}_R^l(\mathbf{p}), \mathbf{F}_F^l(\mathbf{n}) \rangle \,|\, \mathbf{n} \in \mathcal{N}\} \tag{5.1}$$

Finally, a fully convolutional decoder returns the optical flow for each level that is used to warp the features to the next level. This results in a compact and discriminative representation producing state-of-the-art performance.

Inspired by the supervised two-frame PWC-Net model, we develop our unsupervised multi-frame and occlusion aware formulation illustrated in Fig. 5.3. Similar to PWC-Net, we estimate the flow fields and occlusion maps in a coarse-to-fine manner. The first modification we make is to add the past frame to the image and feature pyramids ($\mathbf{F}_P^l$). In the original PWC-Net, a cost volume is constructed based on the features of the reference frame and the features of the future frame warped according to the flow estimate. In contrast, we construct two cost volumes: one for the past and one for the future frame. The two separate cost volumes allow our network to detect occlusions and choose the relevant information for accurate optical flow estimation. Finally, we use three separate decoders for future flow, past flow, and occlusion map, respectively. The cost volumes are stacked together and form the input to the decoders. We upsample past flow, future flow, and occlusion map predictions from the previous level and provide them accordingly as input to the decoders together with the cost volume and the features of the reference frame. The original PWC-Net uses transposed convolutions for upscaling. However, transposed convolutions have been shown to introduce checkerboard artifacts and amplify activations [ODO16; Woj+17; Ran+19b]. Therefore, we instead rely on simple bilinear interpolation for upsampling. For all three decoders, we use the decoder architecture proposed in [Sun+18b] consisting of 5 convolutional layers with 128, 128, 96, 64, and 32 feature channels; just for the occlusion decoder, we add a softmax at the end.

Our architecture with two flow decoders is designed to encourage constant velocity as a soft constraint. We also experiment with an architecture using one flow decoder for both directions. In that case, the inverse future flow is treated as the estimation for past flow. This corresponds to a hard constraint as used in Section 4.2, which is useful in cases where the linear assumption always holds, e.g., on our RoamingImages dataset.

5.3 Loss Functions

Our goal is to learn accurate optical flow and occlusions within a temporal window in an unsupervised manner. Let θ denote the parameters of a neural network, which predicts $\mathbf{U}_F(\theta)$, $\mathbf{U}_P(\theta)$ and $\mathbf{O}(\theta)$ from the input images $\mathcal{I}$. Our loss $\mathcal{L}(\theta)$ is a linear combination of a photometric loss $\mathcal{L}_P(\theta)$, smoothness constraints $\mathcal{L}_{S_P}(\theta), \mathcal{L}_{S_F}(\theta), \mathcal{L}_{S_O}(\theta)$, a constant velocity constraint $\mathcal{L}_{CV}(\theta)$ and an occlusion prior $\mathcal{L}_O(\theta)$:

$$\mathcal{L} = \mathcal{L}_P + \mathcal{L}_{S_F} + \mathcal{L}_{S_P} + \mathcal{L}_{S_O} + \mathcal{L}_O + \mathcal{L}_{CV} \tag{5.2}$$

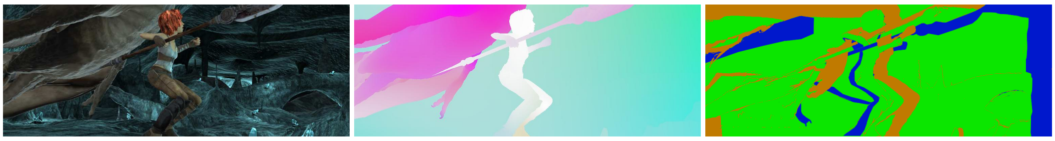
For clarity, we dropped the dependency on the parameters θ and the relative weights of the loss functions. While the first two terms have been frequently employed by unsupervised methods before [PHC16; YHD16; Lon+16; All+17; Vij+17; Ren+17; Wan+18b; MHR18], we extend this formulation to the multi-frame scenario with a simple but effective linear motion model and proper handling of occlusions. In the following, we describe each individual term in detail.

5.3.1 Photometry

In unsupervised optical flow estimation, supervision is achieved by warping the images according to the predicted optical flow and comparing the intensity or color residuals. Unlike existing approaches [PHC16; YHD16; Vij+17; Ren+17; Wan+18b; MHR18], we take advantage of multiple frames to strengthen the photometric constraint. Similar to the works [Wan+18b; MHR18], our model takes occlusions into account. While these methods use simple heuristics based on thresholding to obtain occlusion maps for masking, we directly model occlusions in our formulation and use them to weight the contribution of future and past estimates. Our approach is able to learn more sophisticated models that allow for more accurate occlusion reasoning. Moreover, our approach allows the network to avoid errors in occluded regions since a pixel is, by definition, always visible in at least two frames. More formally, we formulate our photometric loss as

$$\begin{aligned}\mathcal{L}_P = &\sum_{\mathbf{p}\in\Omega} \mathbf{O}^{(2)}(\mathbf{p}) \cdot \delta\left(\mathbf{I}_P\left(\mathbf{p} + \mathbf{u}_P\left(\mathbf{p}\right)\right), \mathbf{I}_R\left(\mathbf{p}\right)\right) \\ &+ \sum_{\mathbf{p}\in\Omega} \mathbf{O}^{(1)}(\mathbf{p}) \cdot \delta\left(\mathbf{I}_F\left(\mathbf{p} + \mathbf{u}_F\left(\mathbf{p}\right)\right), \mathbf{I}_R\left(\mathbf{p}\right)\right)\end{aligned} \tag{5.3}$$

where Ω denotes the domain of the reference image $\mathbf{I}_R$, $\mathbf{u}_P$, and $\mathbf{u}_F$ denote the past and future flow at pixel $\mathbf{p}$, and $\mathbf{O}^{(i)}(\mathbf{p})$ denotes the i'th component of occlusion variable $\mathbf{O}(\mathbf{p})$. Instead of handling occlusions in the warping function, we instead use bilinear interpolation for warping [Jad+15] and a robust function $\delta(\cdot,\cdot)$, detailed below, to measure the photometric error between the warped images $\mathbf{I}'_{P/F} = \mathbf{I}_{P/F}\left(\mathbf{p} + \mathbf{u}_{P/F}\left(\mathbf{p}\right)\right)$ and the reference image $\mathbf{I}_R$. Afterwards, we use our occlusion estimates to weight the photometric errors accordingly. If a pixel $\mathbf{p}$ is more likely to be occluded in the future, $\mathbf{O}^{(1)}(\mathbf{p}) < \mathbf{O}^{(2)}(\mathbf{p})$, the information from the past frame has a larger contribution. Similarly, if a pixel $\mathbf{p}$ is likely occluded in the past, $\mathbf{O}^{(1)}(\mathbf{p}) > \mathbf{O}^{(2)}(\mathbf{p})$, the future frame is weighted higher. In the case of pixel $\mathbf{p}$ being



(a) Reference *(b) Optical Flow* *(c) Occlusions*

Figure 5.4: **Regularization.** *Additional constraints allow us to handle ambiguities in the photometric loss. We visualize the ground truth flow (b) with the typical color coding and the occlusions (c) with visible pixels in green, future and past occlusions in brown and blue. We observe that the optical flow changes smoothly between neighboring pixels of the same object, while occlusion states are identical in the corresponding regions. Finally, we observe that the visible state is more likely to occur than the others.*

visible within the whole window, $\mathbf{O}^{(1)}(\mathbf{p}) \approx \mathbf{O}^{(2)}(\mathbf{p})$, both future and past frames contribute equally. This soft weighting of the data terms ensures that our photometric loss is fully differentiable.

Several photometric error functions have been proposed in the classical optical flow literature. The most popular is the brightness constancy assumption [HS81], which measures the difference between pixel intensities or colors in Eq. (5.4). Instead of the original quadratic penalty function, we use the generalized Charbonnier penalty ρ [BWS05] for robustness against outliers in Eq. (5.6). In realistic scenes with illumination changes, the brightness constancy assumption is often violated, and instead, a gradient constancy assumption is considered by comparing the gradients of the pixel intensities in Eq. (5.5). Therefore, we use the brightness constancy assumption when training on synthetic data and the gradient constancy assumption when training on KITTI.

$$\delta_{BC}(I_1, I_2) = \rho\left(I_1 - I_2\right) \tag{5.4}$$

$$\delta_{GC}(I_1, I_2) = \rho\left(\frac{\partial I_1}{\partial x} - \frac{\partial I_2}{\partial x}\right) + \rho\left(\frac{\partial I_1}{\partial y} - \frac{\partial I_2}{\partial y}\right) \tag{5.5}$$

$$\rho(\mathbf{x}) = \sum_i \sqrt{x_i^2 + 0.001^2} \tag{5.6}$$

5.3.2 Regularization

As discussed in Chapter 3, the photometric loss alone does not sufficiently constrain the problem due to the aperture problem and the ambiguity of local appearance. Thus, we add additional regularizers that encourage smooth flow fields and regions with consistent occlusion states, favors the visible state, and enforces a constant velocity.

Smoothness: The optical flow changes smoothly in image regions corresponding to the same object, as can be seen in Fig. 5.4 (b). Thus similar to Section 4.2.3, we use the following edge-aware smoothness loss for $\mathbf{U}_P$:

$$\mathcal{L}_{S_P} = \sum_{\mathbf{p}\in\Omega} \xi\left(\nabla_x \mathbf{I}_R(\mathbf{p})\right) \rho\left(\nabla_x \mathbf{U}_P(\mathbf{p})\right) + \sum_{\mathbf{p}\in\Omega} \xi\left(\nabla_y \mathbf{I}_R(\mathbf{p})\right) \rho\left(\nabla_y \mathbf{U}_P(\mathbf{p})\right) \tag{5.7}$$

where $\xi(x) = \exp(-\kappa\|x\|_2)$ with $\kappa = 20$ is a contrast sensitive weight to reduce the effect of the smoothness assumption at image boundaries, $\nabla_x \mathbf{I}(x,y) = \mathbf{I}(x,y) - \mathbf{I}(x-1,y)$ and $\nabla_x \mathbf{U}$, accordingly, are the backward difference of the image and flow field in spatial direction x. Following the works [MHR18; Wan+18b], we can replace the first-order smoothness Eq. (5.7) by a second-order smoothness, which allows piecewise affine flow fields when training on KITTI [GLU12]:

$$\begin{aligned}\mathcal{L}_{S_P} = &\sum_{\mathbf{p}\in\Omega} \xi\left(\nabla_x \mathbf{I}_R(\mathbf{p})\right) \xi\left(\Delta_x \mathbf{I}_R(\mathbf{p})\right) \rho\left(\nabla_x \mathbf{U}_P(\mathbf{p}) - \Delta_x \mathbf{U}_P(\mathbf{p})\right) \\ &+ \sum_{\mathbf{p}\in\Omega} \xi\left(\nabla_y \mathbf{I}_R(\mathbf{p})\right) \xi\left(\Delta_y \mathbf{I}_R(\mathbf{p})\right) \rho\left(\nabla_y \mathbf{U}_P(\mathbf{p}) - \Delta_y \mathbf{U}_P(\mathbf{p})\right),\end{aligned} \tag{5.8}$$

Here, $\Delta_x \mathbf{I}(x,y) = \mathbf{I}(x+1,y) - \mathbf{I}(x,y)$ and $\Delta_x \mathbf{U}$, accordingly, denote the forward differences in direction x. The smoothness for the future flow $\mathcal{L}_{S_F}$ is defined accordingly.

Consistent Occlusions: Additionally, we introduce a regularizer, which encourages similar occlusion states at neighboring pixels:

$$\mathcal{L}_{S_O} = \sum_{\mathbf{p}\in\Omega} \xi\left(\nabla_x \mathbf{I}_R(\mathbf{p})\right) \|\nabla_x \mathbf{O}(\mathbf{p})\|_2^2 + \sum_{\mathbf{p}\in\Omega} \xi\left(\nabla_y \mathbf{I}_R(\mathbf{p})\right) \|\nabla_y \mathbf{O}(\mathbf{p})\|_2^2 \tag{5.9}$$

In contrast to the optical flow, the occlusions state is consistent in corresponding regions and only changes abruptly between these regions, as can be observed in Fig. 5.4. The contrast sensitive weighting allows abrupt changes between regions with different occlusion state.

Occlusion Prior: In Fig. 5.4 (c), we can also observe that the majority of pixels are typically visible in all three frames, while occlusions only occur at motion boundaries. Therefore, we introduce a prior that favors the visible state over the occlusion states. We encode this prior as follows:

$$\mathcal{L}_O = -\sum_{\mathbf{p}\in\Omega} \mathbf{O}^{(1)}(\mathbf{p}) \cdot \mathbf{O}^{(2)}(\mathbf{p}) \tag{5.10}$$

Note that Eq. (5.10) is minimized when all pixels are visible (i.e., $\mathbf{O}(\mathbf{p}) = (0.5, 0.5)$).

Constant Velocity: The photometric term and the previous constraints treat the future and past flow separately. In the multi-frame setup, we can go one step further and assume a linear motion model that corresponds to pixels moving with constant velocity within the short temporal window. Despite its simplicity, constant velocity provides a reliable source of information in case of occlusions in addition to spatial smoothness constraints. Under this assumption, the future and past flow should be equal in length but differ in direction. We can enforce this assumption with a hard constraint by predicting only one flow field, as explained in Section 5.2, or as a soft constraint with a future and past flow field. The soft constraint is enforced using a constant velocity loss that we formulate as follow:

$$\mathcal{L}_{CV} = \sum_{\mathbf{p}\in\Omega} \rho\left(\mathbf{U}_P(\mathbf{p}) + \mathbf{U}_F(\mathbf{p})\right) \tag{5.11}$$

5.4 Experimental Results

In this section, we analyze our approach in ablation studies showing the advantages of the multi-frame formulation, occlusion reasoning, and constant velocity assumption for unsupervised learning. In addition, we compare our method to other unsupervised and supervised methods on established optical flow datasets.

We train our network end-to-end using ADAM [KB15] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and weight decay for regularization. We use a batch size of 8 and start with a learning rate of $1e-4$ for pre-training and $1e-5$ for fine-tuning. We pre-train our models for 700K iterations by halving the learning rate after every 200K iteration. For training, we do not use data augmentation because of the large size of RoamingImages.

For evaluation, we consider three standard metrics:

- **Average End-point Error (EPE)**, as introduced in Section 4.4.
- **Average Percentage of Bad Pixels** based on a threshold, i.e., outlier ratio, which is used for evaluation on the KITTI 2015 test set.
- **F1-Score** also introduced in Section 4.4.

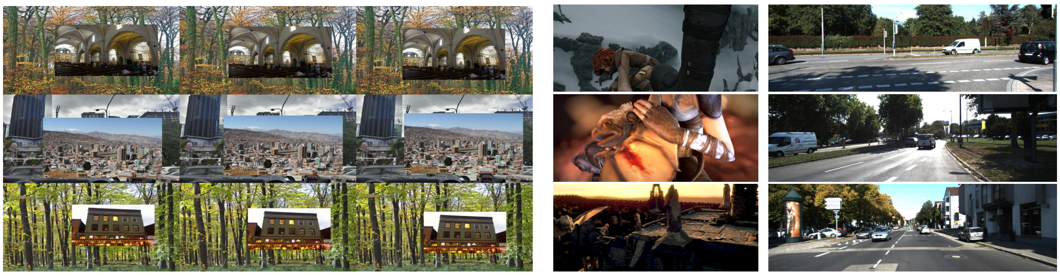
5.4.1 Parameter Settings

Following the original PWC-Net model [Sun+18b], we weight the loss function at each level according to the number of pixels, $[0.005, 0.01, 0.02, 0.08, 0.32]$, and scale flow values by 0.05 as in [Dos+15; Sun+18b].

We use different parameters for training our model on synthetic and real data. For pre-training on RoamingImages and unsupervised fine-tuning on MPI Sintel [But+12] and KITTI 2015 [Gei+13; MG15], we set the hyper-parameters as shown in Table 5.1. The columns, except for the last two, correspond to the relative weights of different terms in the loss function $\mathcal{L}(\theta)$, as defined in (5.2). In particular, those are the parameters of the photometric loss (ω_P), smoothness constraints (ω_{S1}, ω_{S2}), the occlusion prior (ω_O), and the

Table 5.1: **Parameter Settings:** *In this table, we list the dataset specific hyper-parameters that are used in our experiments: the relative weights of the loss functions in the first five columns, the photometric error function as BC (Brightness Constancy) and GC (Gradient Constancy) in the second to the last column, and the order of the smoothness loss in the last column. Each row corresponds to a dataset.*

Dataset	ω_P	ω_{S1}	ω_{S2}	ω_{CV}	ω_O	$\delta(\cdot,\cdot)$	$\mathcal{L}_S$
RoamingImages	2					BC	1st
MPI Sintel	4	0.1	0.1	0.0001	0.1	BC	1st
KITTI 2015	4					GC	2nd



(a) RoamingImages *(b) MPI Sintel* *(c) KITTI 2015*

Figure 5.5: **Datasets.** *The three datasets used in our experiments. For our ablation study and pre-training, we created the synthetic dataset RoamingImages. Eventually, the public datasets KITTI 2015 dataset [Gei+13; MG15] and MPI Sintel [But+12] allow a fair comparison to state-of-the-art approaches.*

constant velocity constraint (ω_{CV}). We use the same parameters for the Clean and Final passes of MPI Sintel.

The column second to the last in Table 5.1 shows the photometric error function used for the dataset. While the brightness constancy (BC) assumption works well with synthetic data (RoamingImages and MPI Sintel), we utilize the gradient constancy (GC) assumption when training on KITTI since it is more robust to illumination changes, which often occur on KITTI.

Finally, we show the order of the smoothness function $\mathcal{L}_S$ in the last column, as mentioned in Section 5.3.2. We use first-order smoothness constraints (1st) on RoamingImages and Sintel, and second-order smoothness constraints (2nd) on KITTI 2015. The second-order smoothness constraint allows piecewise affine flow fields better suited to handle non-fronto-parallel surfaces such as the road region in KITTI.

5.4.2 Datasets

We use three different datasets in our experiments shown in Fig. 5.5. We created a simple dataset called "RoamingImages" to pre-train our model and perform ablation studies. For comparison to other methods, we use two established optical flow datasets in an unsupervised setting, the KITTI 2015 dataset [Gei+13; MG15] and MPI Sintel [But+12].

RoamingImages: Curriculum learning has proven important when training deep models for optical flow estimation [May+18; YHD16; Ilg+17; RB16]. While deep learning approaches for optical flow typically use the FlyingChairs dataset [Dos+15], our multi-frame formulation cannot be trained on this dataset as it provides only two frames per scene. Thus, we have created our own "RoamingImages" dataset by moving a random foreground image in front of a random background image according to random linear motion, as illustrated in Fig. 5.5. The goal is to gradually learn temporal and occlusion relationships by keeping the geometric relations simple in the beginning. We created 80,000 examples with a resolution of 640x320 that we split into 90% training set and 10% test set.

Table 5.2: **Ablation Study:** *We compare our results (Ours) to PWC-Net (Classic) and the multi-frame extension without occlusions (Multi). In addition, we analyze the effect of the constant velocity assumption by turning it off (Ours-None), using a soft constraint (Ours-Soft) or a hard constraint (Ours-Hard). We report flow results using EPE for all (All), non occluded (NOC), and occluded (OCC) pixels.*

Method	Frames	Occlusions	Constant Velocity	All	NOC	OCC	F1
Classic	2	✗	✗	14.14	9.07	32.03	-
Multi	3	✗	hard	10.11	8.24	18.22	-
Ours-None	3	✓	✗	8.37	6.47	16.26	0.76
Ours-Soft	3	✓	soft	8.17	**6.32**	15.87	0.76
Ours-Hard	3	✓	hard	**6.93**	6.89	**8.55**	**0.83**

MPI Sintel: The MPI Sintel dataset [But+12] was introduced in Section 3.7.1 and is one of the most popular datasets for the fair comparison of methods. MPI Sintel provides ground truth flow and occlusion masks for 1000 image pairs in the training set. Two different rendering passes with different complexity are available ("Clean" and "Final") . In addition, MPI Sintel provides pixel-wise occlusion masks. The evaluation of a multi-frame approach is problematic on MPI Sintel since the first frame of every sequence is missing a past frame. We apply our approach on these frames by using the reference frame twice, as reference and past frame.

KITTI 2015: In contrast to MPI Sintel, the KITTI 2015 dataset [Gei+13; MG15] introduced in Section 3.7 provides real scenes for a fair comparison on an evaluation server. While the optical flow training set contains only 200 annotated images, the multi-view extension consists of approximately 4000 images. We use all frames except the annotated frames and their neighbors in the training set (frames 9-12) for unsupervised fine-tuning of our model. We will refer to this set as 'KITTI 2015 MV'. For evaluation, they provide sparse optical flow ground truth obtained using a laser scanner. The occlusions masks only contain regions moving out of the image and, thus, ground truth for occlusions insides the image is not available.

5.4.3 Ablation Study

In this section, we analyze different aspects of our approach on the RoamingImages dataset. More specifically, our goal is to investigate the benefits of our multi-frame formulation with occlusions in comparison to the two-frame case as well as the multi-frame case without occlusion reasoning. In addition, we compare the hard constraint to the soft constraint as well as to the case without any temporal constraints. We list our results in Table 5.2 and discuss our findings in the next paragraph. To perform all experiments, we reduced the number of iterations during training to 300K iterations, which already shows significant differences.

Multi-Frame and Occlusion Reasoning: We first analyze the importance of the multi-frame assumption by training the original two-frame PWC-Net in an unsupervised fashion on RoamingImages (Classic). We then extend PWC-Net to three frames but using only one cost volume without occlusion reasoning (Multi). The multi-frame formulation leads to a significant improvement in the performance, reducing the overall EPE from 14.14 to 10.11 (see Table 5.2). With the multi-frame formulation, even without occlusion reasoning, the error in occluded regions is almost reduced by half. This confirms our motivation for incorporating more information over multiple frames. The occlusion reasoning (Ours-Hard) again reduces the error in occluded regions by half compared to the multi-frame formulation without occlusion reasoning (Multi), reaching an overall EPE of 6.93. This clearly shows the benefit of ignoring misleading information in accordance with the occlusion estimates.

Constant Velocity: As explained in Section 5.3.2, the constant velocity assumption can be enforced in different ways with varying degrees of freedom. In Table 5.2, we compare the soft constraint case (Ours-Soft) with separate flow fields for future and past optical flow, to the hard constraint case (Ours-Hard) with only one flow estimate for both. In addition, we show results without temporal constraint (Ours-None), i.e., turning off the constant velocity term in the loss while still estimating two flow fields. As evidenced by our results, the hard constraint achieves a significant improvement over the case without temporal constraint on our RoamingImages dataset. In particular, in occluded regions, the error is reduced from 16.26 to 8.55 EPE, demonstrating the advantage of the proposed temporal smoothness constraint over a purely spatially regularized model. The soft constraint improves only marginally over the case without temporal constraint, demonstrating the benefit of directly encoding the temporal relationship into the model in our restricted scenario.

5.4.4 Analysis of Feature Maps

In this section, we would like to obtain a deeper understanding of the learned models. The ablation studies so far show that the different design choices lead to overall improvements of the model in terms of accuracy. However, we only consider the output of our model and ignore how the network actually does these predictions. Several approaches have been proposed to further analyze the performance of neural networks by visualizing feature maps [Erh+09; Yos+15]. These ideas can be applied to optical flow networks as presented by Ranjan et al. [Ran+19b].

In the same spirit, we show in Fig. 5.6 the feature activations of our unsupervised model (trained on RoamingImages) and supervised PWC-Net (trained on FlyingChairs) given stationary uniform noise as input. The noise is highly textured and should therefore be easy to match between the images. However, since the noise is stationary, the motion should be zero everywhere. Interestingly, PWC-Net and our unsupervised model show different behavior in intermediate layers even though they share similarities in the architecture. While PWC-Net predicts large motion magnitudes on lower scales (flow6-flow4), our model correctly predicts almost no motion in these cases. Both models are trained with losses on each scale to ensure faster and better training. However, it seems that our unsupervised multi-frame guidance learns a better model on the lower scales than PWC-Net. Another

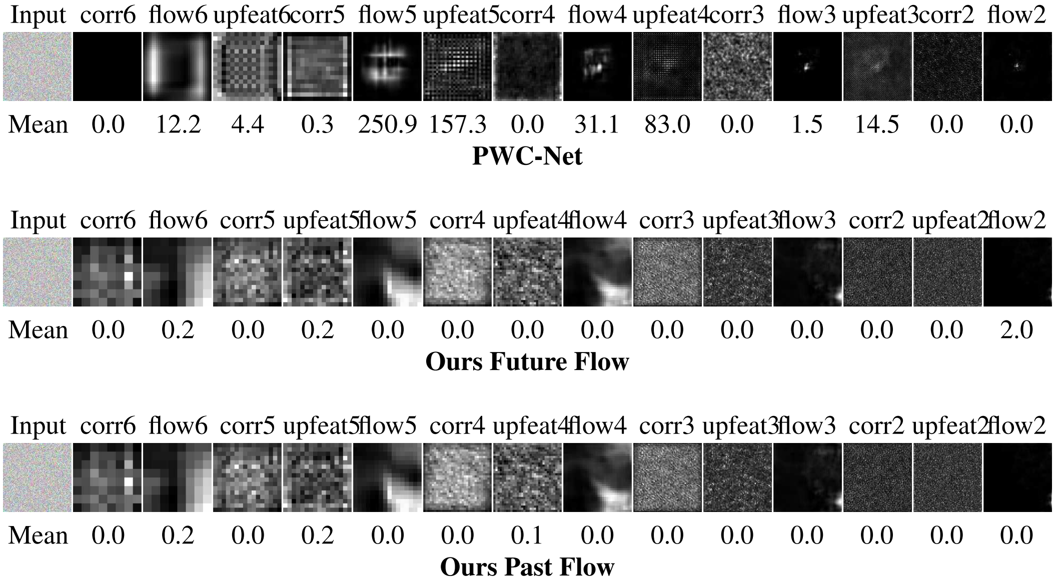

Input corr6 flow6 upfeat6corr5 flow5 upfeat5corr4 flow4 upfeat4corr3 flow3 upfeat3corr2 flow2
Mean 0.0 12.2 4.4 0.3 250.9 157.3 0.0 31.1 83.0 0.0 1.5 14.5 0.0 0.0
PWC-Net
Input corr6 flow6 corr5 upfeat5flow5 corr4 upfeat4flow4 corr3 upfeat3flow3 corr2 upfeat2flow2
Mean 0.0 0.2 0.0 0.2 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 2.0
Ours Future Flow
Input corr6 flow6 corr5 upfeat5flow5 corr4 upfeat4flow4 corr3 upfeat3flow3 corr2 upfeat2flow2
Mean 0.0 0.2 0.0 0.2 0.0 0.0 0.1 0.0 0.0 0.0 0.0 0.0 0.0 0.0
Ours Past Flow

Figure 5.6: Visualization of Feature Maps. *Average norm of the feature maps of supervised PWC-Net and our unsupervised model considering static uniform noise. The images are normalized independently and rescaled to the same size. In addition, we report the average norm of the feature maps (Mean).*

problem can be observed in the transposed convolutions (upfeat6, upfeat5, upfeat4, upfeat3) used by PWC-Net as mentioned in Section 5.2. We can observe checkerboard artifacts in these layers as indicated in [ODO16; Woj+17] and an amplification of the feature activations when considering the mean. By replacing the transposed convolutions with simple bilinear interpolation of the flow fields, our model does not suffer from this problem anymore and has low activations in all layers. One problem that both of the models have in common are the spatial varying activations in the flow layers even though the motion is the same in all locations. Considering the matching problem, the model should learn spatial invariance and give the similar flow estimates for all pixels in this stationary scenario.

5.4.5 Quantitative and Qualitative Results

In Table 5.3, we compare our method to the state-of-the-art unsupervised approaches DSTFlow [Ren+17], UnFlow [MHR18] and OccAwareFlow [Wan+18b], as well as the leading supervised approaches FlowNet [Dos+15], SPyNet [RB16], FlowNet2 [Ilg+17], and PWC-Net [Sun+18b] on MPI Sintel and KITTI 2015 (training and test set). In addition, we show qualitative results on KITTI 2015 and MPI Sintel in Figures 5.7, 5.8, 5.9, 5.10.

While the constant velocity hard constraint works well on the simplistic RoamingImages dataset, more realistic datasets like MPI Sintel and KITTI often exhibit non-linear motions, which violate the constant velocity assumption. Therefore, we exploit the soft constraint network on these datasets initialized based on the hard constraint network pre-trained on

Table 5.3: **Quantitative Results:** *In these tables, we compare our method to several state-of-the-art supervised and unsupervised methods on the training sets (a) and test sets (b) of MPI Sintel and KITTI 2015 datasets. We report the EPE for all (All), non occluded (NOC) and occluded (OCC) pixels except for the KITTI test set where we report the error ratio for all pixels (All) and non-occluded pixels (NOC). Parentheses indicate cases where training was performed on the same dataset and ∗ marks cases where only the annotated samples were excluded from training. Missing entries (-) were not reported for the respective method and bold fonts highlight the best results among supervised and unsupervised methods.*

	Methods	MPI Sintel Clean			MPI Sintel Final			KITTI 2015		
		All	NOC	OCC	All	NOC	OCC	All	NOC	OCC
Supervised	FlowNetS [Dos+15]	4.5	-	-	5.45	-	-	-	-	-
	FlowNetS-ft [Dos+15]	(3.66)	-	-	(4.44)	-	-	-	-	-
	SPyNet [RB16]	4.12	-	-	5.57	-	-	-	-	-
	SPyNet-ft [RB16]	(3.17)	-	-	(4.32)	-	-	-	-	-
	FlowNet2 [Ilg+17]	2.02	-	-	3.14	-	-	10.06	-	-
	FlowNet2-ft [Ilg+17]	**(1.45)**	-	-	**(2.01)**	-	-	(2.3)	-	-
	PWC-Net [Sun+18b]	2.55	-	-	3.93	-	-	10.35	-	-
	PWC-Net-ft [Sun+18b]	(1.70)	-	-	(2.21)	-	-	**(2.16)**	-	-
Unsupervised	DSTFlow [Ren+17]	6.93	5.05	-	7.82	5.97	-	24.3	14.23	-
	DSTFlow-ft [Ren+17]	(6.16)	(4.17)	-	(6.81)	(4.91)	-	16.79*	6.96*	-
	UnFlow-CSS [MHR18]	-	-	-	7.91	-	-	8.10*	-	-
	OccAwareFlow [Wan+18b]	5.23	-	-	6.34	-	-	21.3	-	-
	OccAwareFlow-ft [Wan+18b]	(4.03)	-	-	(5.95)	-	-	8.88*	-	-
	UnFlow-CSS (R) [MHR18]	8.91	-	-	10.01	-	-	19.26	11.44	-
	Ours-Hard	5.38	4.32	11.58	6.01	4.92	**12.42**	15.63	8.8	41.65
	Ours-Hard-ft	(6.05)	(4.95)	(12.10)	(7.09)	(5.97)	(13.42)	11.58*	7.45*	27.29*
	Ours-None-ft	(4.74)	(3.60)	(11.42)	(5.84)	(4.72)	(12.66)	6.65*	**3.24***	19.33*
	Ours-Soft-ft-Kitti	5.69	4.52	12.68	6.48	5.33	13.46	**6.65***	3.42*	**18.51***
	Ours-Soft-ft-Sintel	**(3.89)**	**(2.64)**	**(11.21)**	**(5.52)**	**(4.32)**	(12.87)	15.69	7.87	46.34

***(a)** Training*

	Methods	MPI Sintel Clean			MPI Sintel Final			KITTI 2015	
		All	NOC	OCC	All	NOC	OCC	All	NOC
Supervised	FlowNetS-ft [Dos+15]	6.69	-	-	7.46	-	-	-	-
	SPyNet-ft [RB16]	6.64	3.01	36.19	8.36	4.51	39.69	35.07%	26.71%
	FlowNet2-ft [Ilg+17]	3.60	1.46	**24.30**	5.74	2.75	30.11	11.48%	6.94%
	PWC-Net-ft [Sun+18b]	**3.86**	**1.45**	26.17	**5.17**	**2.44**	**26.22**	**9.60%**	**6.12%**
Unsupervised	DSTFlow [Ren+17]	10.4	5.2	-	11.11	5.92	-	-	-
	DSTFlow-ft-Kitti [Ren+17]	10.95	5.87	-	11.8	6.7	-	39.00%	-
	DSTFlow-ft-Sintel [Ren+17]	10.41	5.3	-	11.28	6.16	-	-	-
	UnFlow-CSS [MHR18]	9.38	5.37	42.16	10.22	6.06	44.11	23.30%	14.68%
	OccAwareFlow [Wan+18b]	8.02	-	-	9.08	-	-	-	-
	OccAwareFlow-ft [Wan+18b]	7.95	4.08	39.53	9.15	5.21	41.31	31.20%*	23.53%*
	Ours-Hard	8.35	4.81	37.14	9.38	5.76	**38.84**	48.93%	41.09%
	Ours-Soft-ft	**7.23**	**3.60**	**36.78**	**8.81**	**5.03**	39.65	**22.94%**	**13.85%**

***(b)** Test*

RoamingImages. More specifically, we copy the parameters of the flow decoder in the pre-trained network to the future and past flow decoders while inverting the sign of the past flow decoder's output. We empirically found this to yield a good initialization for further fine-tuning. Afterwards, we fine-tune our model on the target datasets, i.e., KITTI 2015 MV and MPI Sintel. Note that, during fine-tuning, the model is still trained in an unsupervised fashion. In the following, we present our results in comparison to several state-of-the-art approaches.

Pre-training: Since fine-tuning on a specific dataset makes a big difference, we first consider unsupervised methods without fine-tuning to evaluate our pre-trained model on RoamingImages. Our pre-trained model (Ours-Hard) achieves comparable results on MPI Sintel Clean and significantly outperforms all other unsupervised models without fine-tuning on MPI Sintel Final and KITTI 2015. While the best EPE obtained by a pre-trained unsupervised model is 6.34 on MPI Sintel Final training and 21.30 on KITTI 2015 training (Table 5.4a), our model achieves an EPE of 6.01 and 15.63, respectively. On MPI Sintel Final, we are even on par with the model of OccAwareFlow [Wan+18b] fine-tuned on MPI Sintel. This is particularly impressive considering the simplistic dataset used for training our model consisting of linear motions and rectangular images. We observe similar results on the test set of both datasets in Table 5.4b.

Hard vs. Soft Constraint: We compare our hard constraint network to our soft constraint variant to demonstrate the necessity to relax the constant velocity assumption for more complex datasets. While our model with hard constraint (Ours-Hard-ft) improves after fine-tuning on KITTI 2015 training, its performance is still behind other unsupervised, fine-tuned approaches. On MPI Sintel, the performance decreases after fine-tuning because the constant velocity constraint is wrongly enforced on non-linear motion, which frequently occurs in this dataset. Switching to the soft constraint version (Ours-Soft-ft) allows deviations from constant velocity assumption and results in significant improvements on both datasets. For completeness, we include our fine-tuned model without temporal constraint (Ours-None-ft) in the comparison on the training sets. Similar to Table 5.2, the performance of the model without temporal constraint (Ours-None-ft) is inferior to the one with the soft constraint (Ours-Soft-ft) in all cases except the occluded regions (OCC) on MPI Sintel Final and not occluded regions (NOC) on KITTI 2015. On KITTI 2015, the improvements in the occluded regions are marginal due to dominating complex motions. We conclude that fine-tuning with the soft constraint is in general beneficial even when complex motions violate the constant velocity assumption.

Results with Fine-tuning: The performance significantly improves in non-occluded regions on all datasets after fine-tuning using the soft constraint (Ours-Soft-ft). In occluded regions, there are only minor improvements or even a degradation in performance (Sintel Final). The soft constraint allows deviations from the constant velocity model resulting in improvements in non-occluded regions with complex motion. However, less temporal information is available for occluded regions when switching from the hard-constraint to the soft-constraint. In other words, the predictions rely more on spatial information than on temporal information. Still, the overall performance improves with the soft-constraint since

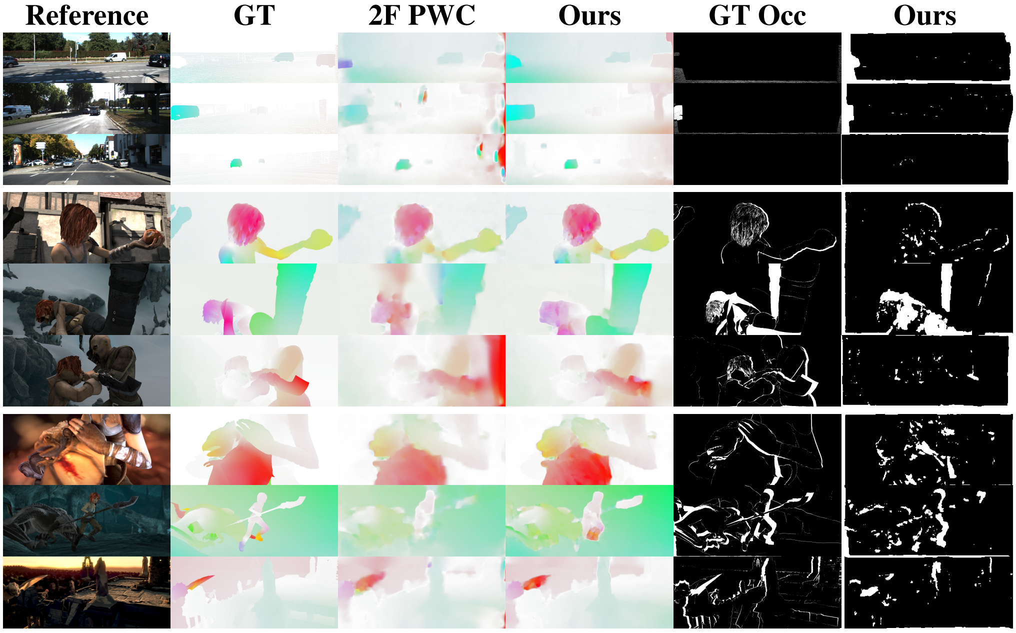


Figure 5.7: **Qualitative Comparison:** *We compare our final results (fourth column) to two-frame PWC-Net (third column) on examples from KITTI 2015 (upper three rows) and MPI Sintel Clean (middle three rows) and MPI Sintel Final (bottom three rows). Our model produces better flow estimates with sharper boundaries as well as accurate occlusion estimates (last column).*

non-occluded regions typically cover a larger area compared to occluded regions.

Our soft constraint model fine-tuned on MPI Sintel (Ours-Soft-ft) achieves an EPE of 3.89 and 5.52 on Clean and Final training, hence outperforming all other unsupervised methods while even achieving comparable results to FlowNet fine-tuned on MPI Sintel Clean. Similarly, on the test set, we outperform all other unsupervised methods with 7.23 and 8.81 EPE on Clean and Final, performing on par with supervised methods without fine-tuning, e.g., FlowNet and SPyNet. Fine-tuning on KITTI 2015 MV improves the performance to 6.59 in comparison to 8.10, the best-achieved EPE by an unsupervised method so far. On the test set, we even achieve better performance than UnFlow that is trained on a large synthetic dataset (Synthia [Ilg+17]) and KITTI Raw dataset.

Fig. 5.7 shows a qualitative comparison of our fine-tuned models on each dataset to the two-frame formulation. Our multi-frame formulation with occlusions results in more accurate optical flow fields with sharp motion discontinuities as well as occlusion estimates. We show additional qualitative results of our fine-tuned models on KITTI 2015 (Fig. 5.8), MPI Sintel Clean (Fig. 5.9), and Final (Fig. 5.10). Despite missing explicit supervision, our models yield accurate and sharp optical flow predictions. However, large motions and fine details can lead to some failure cases, as in the last three rows in Fig. 5.8, Fig. 5.9, and Fig. 5.10.

Table 5.5: **Occlusion Estimates:** *We compare the performance of our occlusion estimation to other approaches on MPI Sintel and KITTI 2015 using the F-Measure. Parentheses indicate cases where training was performed on the same dataset while $*$ marks cases where only the annotated samples were excluded from training. Note that S2D [LZS13] is a supervised method.*

Methods	MPI Sintel Clean	MPI Sintel Final	KITTI 2015
S2D [LZS13]	-	**0.57**	-
MODOF [XJM12]	-	**0.48**	-
OccAwareFlow-ft [Wan+18b]	**(0.54)**	**(0.48)**	0.88*
Ours-Soft-ft	(0.49)	(0.44)	**0.91***

Cross-dataset Performance: Table 5.4a also shows the cross-dataset performance of our approach, i.e., trained on one dataset and tested on another, compared to the previous approaches. Our model fine-tuned on KITTI 2015 performs similarly to the pre-trained model on MPI Sintel and vice versa. This shows the generalization capability of our approach without over-fitting to a specific dataset.

Occlusion Estimation: We evaluate our occlusion masks on both MPI Sintel and KITTI 2015 datasets. We compare our results quantitatively to OccAwareFlow [Wan+18b], S2D [LZS13], and MODOF [XJM12] using the F-Measure (Table 5.5). While OccAwareFlow [Wan+18b] obtains occlusion estimations considering the backward flow, S2D [LZS13] uses a binary classification, and MODOF [XJM12] uses a discrete-continuous optimization of an energy function.

With unsupervised fine-tuning on MPI Sintel (Ours-Soft-ft), we obtain comparable results to OccAwareFlow [Wan+18b]. Learning occlusions on MPI Sintel in an unsupervised fashion is very difficult since occlusions often occur in untextured regions with limited guidance by the photometric loss. Even the supervised approach S2D struggles on the MPI Sintel dataset, only reaching an F-Measure of 0.57. Moreover, similar to the original PWC-Net [Sun+18b], we estimate the optical flow and occlusion mask on quarter resolution. While larger occlusions are mostly estimated correctly, fine details are usually missing due to downsampling, as can be observed in the qualitative results (Fig. 5.7). On KITTI 2015, the occlusion masks only contain pixels moving out of the image. Considering these masks, we reach the best performance with our unsupervised fine-tuned model (Outs-Soft-Kitti-ft). Note that several occlusions missing in the ground truth masks are correctly estimated by our method, e.g., the vehicles leaving the image in Fig. 5.7.

Contribution of RoamingImages: In contrast to other unsupervised approaches, we pre-train our model on our RoamingImages dataset since there are no simple multi-frame datasets available. This raises the question of whether the reason for the success of our model is our dataset due to its size, simplicity, or some other factor. To dispel this doubt, we pre-train UnFlow CSS [MHR18] on our dataset and compare its performance to our pre-trained model. We use the code provided with default parameters only by changing the learning rate to $1e-5$. As shown in Table 5.4a, our pre-trained model (Ours-Hard) performs significantly better than UnFlow CSS trained on the same data (UnFlow-CSS (R)) on all datasets. This shows that the success of our approach is not solely based on our new dataset but critically depends on the proposed multi-frame formulation.

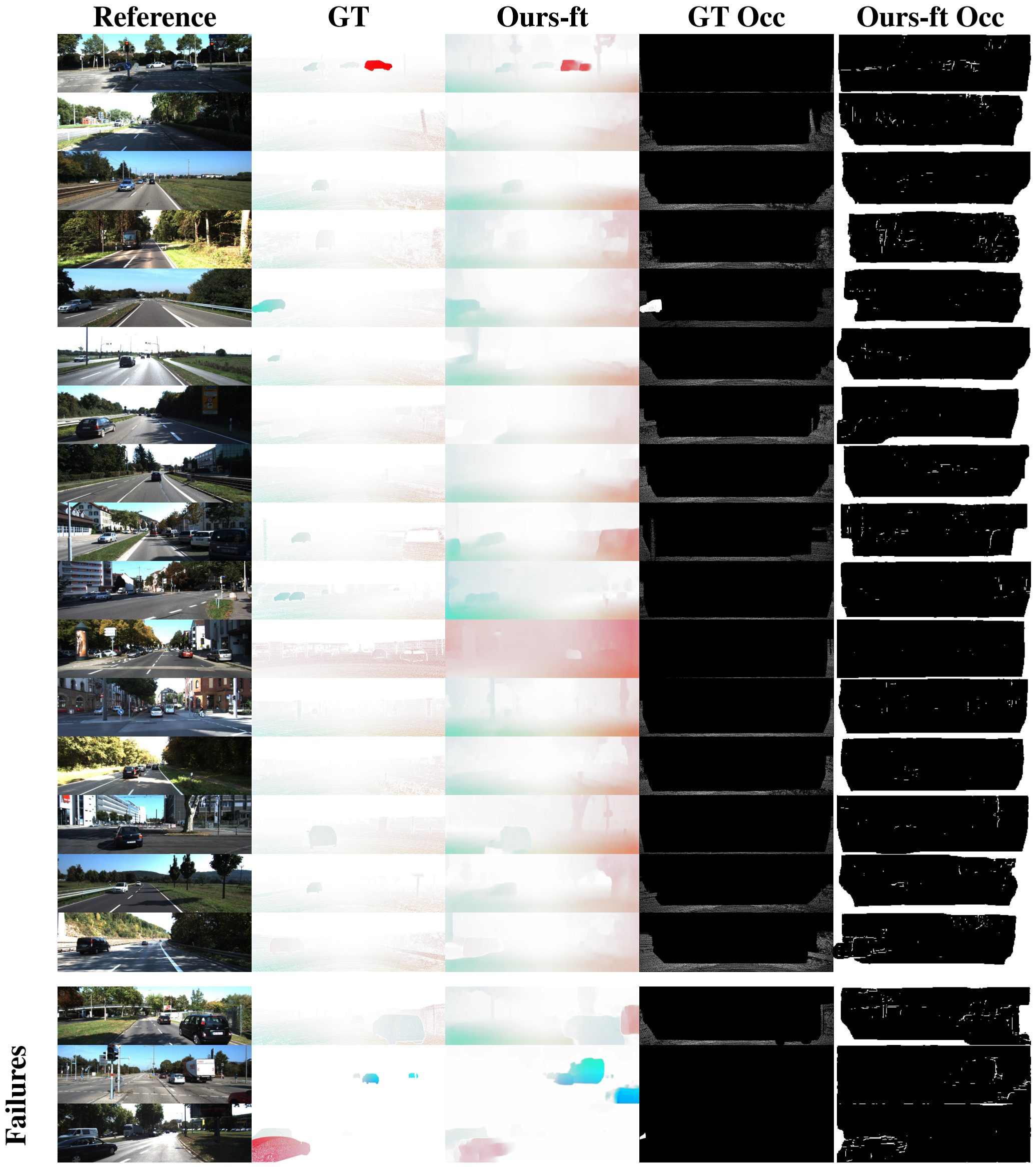


Figure 5.8: **Qualitative Results:** *In this figure, we show our results with multiple frames and occlusion reasoning (third column) on examples from KITTI 2015. Our model produces accurate flow estimates with sharp boundaries as well as accurate occlusion estimates (last column).*

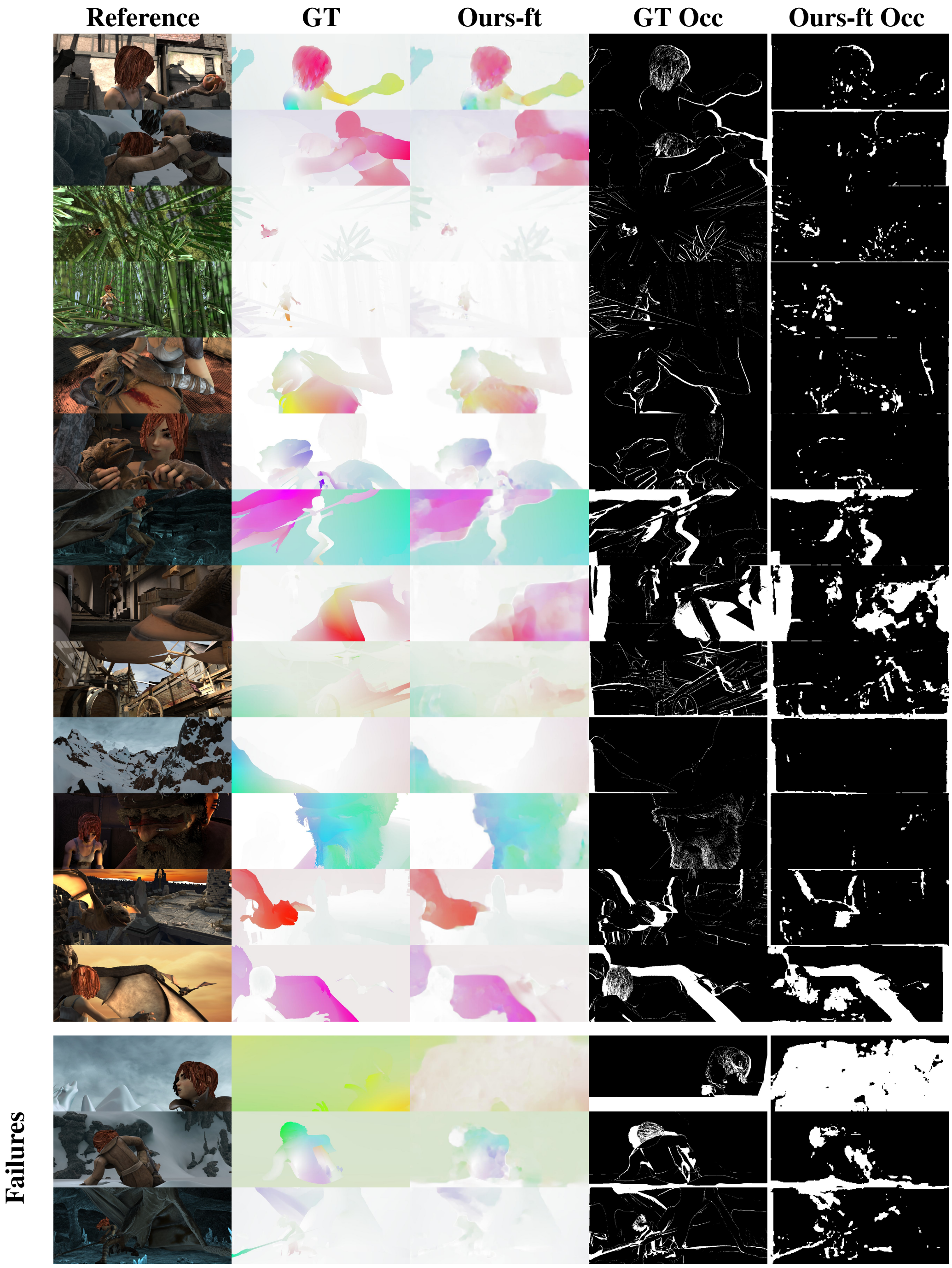


Figure 5.9: **Qualitative Results:** *In this figure, we show our results with multiple frames and occlusion reasoning (third column) on examples from MPI Sintel Clean. Our model produces accurate flow estimates with sharp boundaries as well as accurate occlusion estimates (last column).*

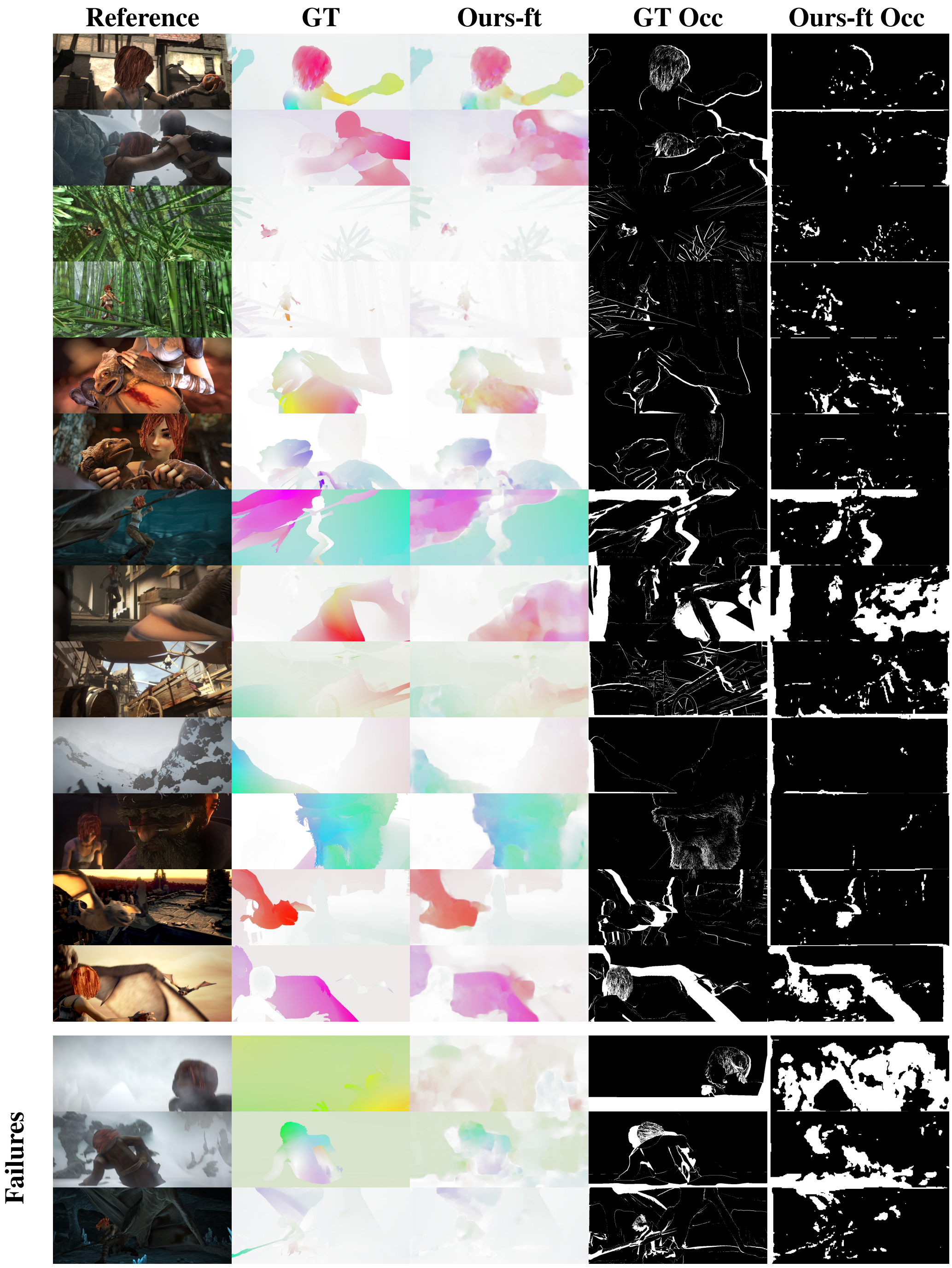


Figure 5.10: **Qualitative Results:** *In this figure, we show our results with multiple frames and occlusion reasoning (third column) on examples from MPI Sintel Final. Our model produces accurate flow estimates with sharp boundaries as well as accurate occlusion estimates (last column).*

6 Conclusion and Outlook

After the first success in 2015[Dos+15], deep neural networks have become the preferred choice when addressing the optical flow problem. However, the limited number of real-world datasets for optical flow poses a problem for the evaluation and training of modern optical flow methods. In this thesis, we investigated two different directions to address this data scarcity problem.

Generating Reference Data: In Chapter 4, we focused on the problem of generating new datasets for the optical flow problem. We leverage the vast amount of temporal information from high-speed cameras to generate reference data by tracking pixel densely through time. We formulate an optimization problem over the large space-time volume and make it feasible by splitting it into simple subproblems. First, we enforce strong temporal constraints while estimating Flowlets, the optical flow between intermediate frames of the HFR videos. We extended an accurate variational method to jointly infer the optical flow and occlusions for a temporal window. Strong temporal constraints in the formulation allowed us also to propagate information through time and improve the estimation in occluded regions. Second, we formulate a dense tracking problem and use the intermediate flow results to establish accurate reference data even with large displacements in the final frame rate. We discretized the solution space and used MP-PBP for optimization with a data-driven initialization of the particles.

We compared our approach to a popular two-frame formulation on an HFR version of the MPI Sintel dataset and several real-world sequences. We conclude that the generated reference data is precise enough for an insightful comparison of optical flow methods. Thus, we created a real-world dataset using our approach with novel challenges for evaluating the state of the art in optical flow. In our comparison, we observed that all methods except FlowNet, SPyNet, FlowNet2, PWCNet and ClassicNL strongly suffer from motion blur. The magnitude of the flow affects in particular FlowNet, SPyNet and variational approaches, which cannot handle large displacements well compared to methods guided by matching or optimizing local feature correspondences. However, the learning-based approaches FlowNet2 and PWCNet outperform all others in all challenges.

While the multi-frame variational and dense tracking formulation achieve impressive results, there is still great potential for improvements. Complex occlusions and untextured regions are still causing errors in the estimation. On the one hand, the reliable detection of occlusions is difficult, even when considering high frame rates. A joint optical flow and occlusion formulation is a good direction since both problems are closely related to each other. While occlusion estimates allow for a better explanation of the observations, the optical flow causes occlusions and can be used to improve the predictions in occluded regions. On the other hand, the estimation in untextured regions does not benefit from the additional temporal information given in HFR sequences since the problem will affect all frames. In this case, we rely only on information from neighboring pixels, and in complex scenes, this information is not sufficient to make accurate predictions.

Both problems cannot be resolved using classical optimization methods with simple assumptions. More complex models are necessary, and a learning-based approach, as investigated in the second part of the thesis, could learn such models directly from the data without annotations. In the same spirit, better motion models can be learned to improve the prediction of the optical flow in occluded regions as suggested by Maurer and Bruhn [MB18]. In addition to the learning-based approach, a layered representation of the scene into independently moving objects and the static scene would be beneficial, as in the works [YMU13; YMU14; Bai+16; Sev+16; WSB17; Vij+17; Ran+19a]. Based on the epipolar geometry between all images, the motion of each pixel in the static scene can be described by the camera motion and the depth. In contrast to optical flow, the camera motion enforces a strong constraint on the optical flow of the entire static scene, and the estimation of the depth reduces to a 1D search problem along the epipolar line. In addition, all images can be leveraged to infer the structure of the scene, since it is constant over time. Occlusions and the motion in occluded regions of the static scene can be accurately estimated, given the camera motion and structure of the scene. Since human-made environments usually consist of many planar surfaces, untextured regions can be addressed with strong constraints on the structure of the scene. The depth estimation problem could be further alleviated with the introduction of a second synchronized high-frame-rate camera mounted at a fixed baseline. Then, the resulting dataset could also be used for stereo matching and scene flow.

Finally, it would be interesting to incorporate a more reliable measure of confidence. Instead of using simple flow consistency or color saturation to filter out bad estimates, a probabilistic extension, as suggested by Gast and Roth [GR18], should be investigated. The uncertainties from a probabilistic approach would allow a much more reliable measure of confidence. The confidences could be taken into account in the evaluation of methods or during training giving smaller weights to uncertain and higher weights to certain estimates.

Unsupervised Learning: While new annotated real-world datasets are from the utmost importance for the evaluation of methods, the generation will always involve some additional labor in addition to the acquisition. The acquisition can be avoided by using HFR videos from the Internet, but the estimation is still computationally demanding because of the continuous and discrete variables over the large space-time volume. In contrast, unsupervised learning discussed in Chapter 5 allows using large Internet collections without any annotations.

While unsupervised learning of optical flow has been investigated before, many methods do not model occlusions and the few modeling occlusions use simple heuristics. In combination with a photometric loss, this leads to strong misleading gradients in occluded regions. Therefore, we presented a method for unsupervised learning of optical flow and occlusions from multiple frames. We proposed modifications to a state-of-the-art two-frame architecture for handling multiple frames in order to predict past and future optical flow as well as an occlusion map within a temporal window. We formulated unsupervised loss functions to exclude misleading information in occluded regions and incorporate a simple temporal model. In the experimental results, we show the benefits of the multi-frame formulation with occlusions over classical two-frame formulations and the importance of directly modeling temporal relations. We achieve impressive results with proper modeling and unsupervised training on a simple dataset outperforming all previous unsupervised methods on complex datasets. After unsupervised fine-tuning, our approach is even on par with some fully supervised methods.

Our results on unsupervised learning highlight the hidden potential of unsupervised methods and motivate more in-depth examinations. However, the occlusion estimates of our formulation are still very rough. Because of memory limitations during training, we had to rely on lower output resolutions for the optical flow and occlusions estimates. This has a large effect on the occlusion maps since the occlusions are in general bulkier, and thin occlusions are lost. This problem can be resolved by considering lighter architectures following similar ideas as SPyNet [RB16] or only estimating the occlusion mask on higher resolution. The estimation of motion in occluded regions is also still very challenging. While temporal information already improved the performance significantly, other sources of supervision should be investigated. For instance, Liu et al. [Liu+19] suggest to train a network in an unsupervised fashion and use the pre-trained model to create new training examples for occluded regions by randomly covering image regions that were visible before. This allows them to generate an arbitrary amount of pseudo ground truth for occluded regions and refine the model on this data. Similar to the previous discussion, it would be interesting to split the scene into independently moving objects and the static scene. While Vijayanarasimhan et al. [Vij+17] and Ranjan et al. [Ran+19a] considered only one frame to infer the structure, a three-frame formulation as we suggest would allow much more accurate depth estimations. In addition, both do not take into account occlusions, which are crucial for unsupervised learning.

Reference Data from Unsupervised Learning: As mentioned before, learning the detection of occlusions and the motion in these regions would be beneficial for the dense tracking formulation. Therefore, a combination of both ideas to generate richer datasets is worth an investigation. A big advantage of the learning-based approaches is efficiency since only a forward pass is necessary. In contrast, the classical optimization methods used in the first part of the thesis are very time-consuming. A more efficient approach allows the generation of larger datasets in a shorter time and would, therefore, be very useful.

One simple way to combine the methods would be to replace the variational estimation with a learned model. However, the learned model would need to achieve subpixel precision on a large resolution while all models so far work on low resolutions due to the limited GPU memory. This problem could be alleviated by approaching the dense tracking problem with a temporal residual estimation problem similar to [LG01; LAG04; LAG05]. Considering a reference frame R and target frame $T = R + S$ with S intermediate frames from a HFR sequence, we start with the motion estimation between R and $\hat{T} = R + 1$ and incrementally increase $\hat{T}$ while using the previous motion estimation for warping. In each iteration, a residual motion estimation problem is addressed, which is much simpler than the motion estimation from R to $\hat{T}$. In addition, accumulated errors (drift) can be reduced by considering new image information. The motion between intermediate HFR frames will be in the magnitude of several pixels and, therefore, a shallow network will be sufficient to estimate the motion similar to the decoder networks used in PWC-Net and SPyNet. Such small networks will meet the restricted memory requirements of GPUs and prevent overfitting. In combination with an LSTM [PHC16], the previous motion estimation can be taken into account to enforce temporal constraints. The memory of the LSTM would be used to store the previous motion estimation and to provide this information for the estimation of the next frames. Since only the Sintel dataset exists for HFR videos with optical flow ground truth, we would need to relate to unsupervised learning or semi-supervised learning.

Deep Learning for Optical Flow: Deep learning has become the first choice when addressing the optical flow problem. In contrast to classical optimization methods, they allow learning high-level representations from data, which constrain the solution space. Another great advantage is the speed during inference. While the time-consuming training is performed offline, the learned model is eventually applied with only one forward pass. In addition, the high parallelization of GPUs can be exploited leading to real-time performance of several optical flow networks [Dos+15; RB16; Ilg+17; Sun+18b].

The efficiency and high accuracy of optical flow networks will lead in the future to even more interest in using correspondences for other problems. Similar to action recognition, many computer vision problems will benefit from such dense correspondences. For instance, only few approaches leverage temporal information in object detection [Zhu+17b; Zhu+17a; Zhu+18; Wan+18a] and semantic segmentation [He+17; Ma+17; KVK16]. In these cases, temporal consistency allows for more reliable systems since objects and semantics are constant over time. The reliability of these systems will become more important in the future since they are applied in safety-critical environments where each error can cause fatalities. In addition, motion patterns are an insightful cue for detecting different objects in a scene and provide important information about the behavior of these objects. All these applications will set new goals and challenges for the optical flow community.

Nevertheless, convolutional neural networks used in the optical flow literature still share some weaknesses. In [Ran+19b], we investigated adversarial attacks on several different architectures for optical flow estimation. Adversarial attacks [Sze+13; GSS14; NYC15] are popular to analyze the reliability of convolutional networks for objection classification. These approaches seek small perturbations of the input that will cause the network to make wrong predictions. Recently, physical attacks [Sha+16; AS17; Evt+17] have been introduced that explore attacks in the physical world. For instance, Brown et al. [Bro+17] suggest learning a patch that will cause the network to predict the wrong class. Such kind of attacks can be easily replicated in the real world by just printing out the patch.

Inspired by these attacks, we optimized a small patch that is inserted into the input images such that the angle between the flow prediction and the ground truth is maximized. We considered encoder-decoder architectures FlowNet [Dos+15] and FlowNet2 [Ilg+17] as well as spatial pyramid networks SPyNet [RB16], PWC-Net [Sun+18b], and our unsupervised method. Spatial pyramid networks turned out to perform better in general under these attacks than encoder-decoder networks. Eventually, we were able to find a successful attack for each network architecture, which indicates that these models might not necessarily learn the concept of motion.

Finally, we visualized the feature activations for all networks given a stationary uniform noise input, as discussed in Section 5.4.4. Our adaption of PWC-Net presented in Section 5.1 resolved the problem of checkerboard artifacts caused by transposed convolutions and showed better behavior on lower scales. However, none of the models were spatial invariant with respect to the matching problem. The activations of the flow layers were varying spatially, while the motion was constant. It seems that spatial invariance cannot be learned with the simple data augmentation of random spatial shifts as all approaches use this kind of data augmentation. Instead, an additional loss could be introduced that enforces spatial invariance during training. In addition, network architectures better suited for the optical flow problem should be investigated by taking into consideration classical ideas. For example, Hur and Roth [HR19] proposed iterative versions of popular networks inspired by classical optimization, which lead to significant improvements. For unsupervised learning, better optimization methods than simple gradient descent during training could be considered since they also led to great improvements in classical approaches [SRB10].

Conclusion: In this thesis, we presented two directions to address the data scarcity problem in optical flow. While there is still high potential to improve modern optical flow methods, a proper evaluation of the methods is of the utmost importance to ensure steady advancements. For this purpose, new challenging real-world benchmarks can be created using our reference data generation approach, as presented in Chapter 4. In addition, new schemes for training these methods without supervision are important for closing the gap between training and testing performance, as discussed in Chapter 5. This thesis provided approaches and important results for both directions, which will hopefully allow the community for further advances in the optical flow problem.

A Publications

Publications [Jan+17b; Jan+18] are covered in this thesis:

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 Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2017

J. Janai, F. Güney, A. Ranjan, M. Black, and A. Geiger. "Unsupervised Learning of Multi-Frame Optical Flow with Occlusions". In: *Proc. of the European Conf. on Computer Vision (ECCV)*. 2018

Other publications during the Ph.D. [Jan+17a; Ran+19b; Liu+20] that are not completely covered in this thesis:

J. Janai, F. Güney, A. Behl, and A. Geiger. "Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art". In: *arXiv.org* 1704.05519 (2017)

A. Ranjan, **J. Janai**, A. Geiger, and M. Black. "Attacking Optical Flow". In: *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*. 2019

P. Liu, **J. Janai**, M. Pollefeys, T. Sattler, and A. Geiger. "Self-Supervised Motion Deblurring". In: *IEEE Robotics and Automation Letters (RA-L)* (2020)

I declare that this thesis has been created by me based on my own original research mentioned above. My advisor Prof. Dr. Andreas Geiger was involved in all projects and my advisor Prof. Dr. Michael Black was involved in the majority of projects. Both contributed ideas and text to my publications. Prof. Dr. Andreas Geiger also contributed illustrative figures. My co-author Dr. Jonas Wulff created the HFR version of Sintel, and my co-author Dr. Fatma Güney also contributed to the code used in the benchmark and unsupervised method. Both also contributed ideas and text to the publications. I also use content from [Jan+17a; Ran+19b] in this thesis. The introduction to the optical flow problem (Chapter 3) is based on the optical flow chapter in Janai et al. [Jan+17a], which was a collaboration with Dr. Fatma Güney, Aseem Behl and Prof. Dr. Andreas Geiger. The feature analysis presented by Dr. Anurag Ranjan and myself [Ran+19b] was used in the ablation study of the unsupervised method (Section 5.4), and the general results of the work were shortly discussed in Chapter 6. The other sources, including data and code, are referenced in the text. All the experiments in this work are the result of my own work unless otherwise stated.

Bibliography

[ACM15] P. Agrawal, J. Carreira, and J. Malik. “Learning to See by Moving”. In: *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*. 2015.

[All+17] S. Alletto, D. Abati, S. Calderara, R. Cucchiara, and L. Rigazio. “TransFlow: Unsupervised Motion Flow by Joint Geometric and Pixel-level Estimation”. In: *arXiv.org* (2017).

[Ama] Amazon. *Amazon Mechanical Turk*. `https://www.mturk.com/`. [Online; accessed 29-October-2019].

[Ana89] P. Anandan. “A computational framework and an algorithm for the measurement of visual motion”. In: *International Journal of Computer Vision (IJCV)* 2.3 (1989), pp. 283–310.

[AS17] A. Athalye and I. Sutskever. “Synthesizing robust adversarial examples”. In: *arXiv.org* (2017).

[BA91] M. J. Black and P. Anandan. “Robust dynamic motion estimation over time”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 1991.

[BA93] M. J. Black and P. Anandan. “A framework for the robust estimation of optical flow”. In: *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*. 1993.

[BA96] M. J. Black and P. Anandan. “The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields”. In: *Computer Vision and Image Understanding (CVIU)* 63.1 (1996), pp. 75–104.

[Bai+16] M. Bai, W. Luo, K. Kundu, and R. Urtasun. “Exploiting Semantic Information and Deep Matching for Optical Flow”. In: *Proc. of the European Conf. on Computer Vision (ECCV)*. 2016.

[Bak+11] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski. “A Database and Evaluation Methodology for Optical Flow”. In: *International Journal of Computer Vision (IJCV)* 92 (2011), pp. 1–31.

[Bar12] D. Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012. ISBN: 0521518148.

[BB87] R. C. Bolles and H. H. Baker. “Epipolar-plane Image Analysis: A Technique for Analyzing Motion Sequences”. In: *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*. Ed. by M. A. Fischler and O. Firschein. 1987.

[Bes+14] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz. “PMBP: PatchMatch Belief Propagation for Correspondence Field Estimation”. In: *International Journal of Computer Vision (IJCV)* 110.1 (2014), pp. 2–13.

[BKP10] K. Bredies, K. Kunisch, and T. Pock. “Total Generalized Variation”. In: *Journal of Imaging Sciences (SIAM)* 3.3 (2010), pp. 492–526.

[BM11] T. Brox and J. Malik. “Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation”. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)* 33 (Mar. 2011), pp. 500–513.

[Bre65] J. E. Bresenham. “Algorithm for Computer Control of a Digital Plotter”. In: *IBM* 4.1 (1965), pp. 25–30.

[Bro+04] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. “High accuracy optical flow estimation based on a theory for warping”. In: *Proc. of the European Conf. on Computer Vision (ECCV)*. 2004.

[Bro+17] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer. “Adversarial patch”. In: *arXiv.org* (2017).

[BTS15] C. Bailer, B. Taetz, and D. Stricker. “Flow Fields: Dense Correspondence Fields for Highly Accurate Large Displacement Optical Flow Estimation”. In: *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*. 2015.

[BTS17] C. Bailer, B. Taetz, and D. Stricker. “Optical Flow Fields: Dense Correspondence Fields for Highly Accurate Large Displacement Optical Flow Estimation”. In: *arXiv.org* abs/1703.02563 (2017).

[But+12] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. “A naturalistic open source movie for optical flow evaluation”. In: *Proc. of the European Conf. on Computer Vision (ECCV)*. 2012.

[BVZ99] Y. Boykov, O. Veksler, and R. Zabih. “Fast Approximate Energy Minimization via Graph Cuts”. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)* 23 (1999), p. 2001.

[BW06] A. Bruhn and J. Weickert. “A Confidence Measure for Variational Optic flow Methods”. In: *Geometric Properties for Incomplete Data*. Ed. by R. Klette, R. Kozera, L. Noakes, and J. Weickert. Dordrecht: Springer Netherlands, 2006, pp. 283–298. ISBN: 978-1-4020-3858-7. DOI: 10.1007/1-4020-3858-8_15.

[BW16] M. Buczko and V. Willert. “Flow-Decoupled Normalized Reprojection Error for Visual Odometry”. In: *Proc. IEEE Conf. on Intelligent Transportation Systems (ITSC)*. 2016.

[BWS05] A. Bruhn, J. Weickert, and C. Schnörr. “Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods”. In: *International Journal of Computer Vision (IJCV)* 61.3 (2005), pp. 211–231.

[Che+14] X. Chen, Z. Qin, L. An, and B. Bhanu. “An Online Learned Elementary Grouping Model for Multi-target Tracking”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[CK16] Q. Chen and V. Koltun. “Full Flow: Optical Flow Estimation By Global Optimization over Regular Grids”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[Cor+16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[DHS11] J. C. Duchi, E. Hazan, and Y. Singer. “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”. In: *Journal of Machine Learning Research (JMLR)* 12 (2011), pp. 2121–2159.

[Dos+15] A. Dosovitskiy, P. Fischer, E. Ilg, P. Haeusser, C. Hazirbas, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. “FlowNet: Learning Optical Flow with Convolutional Networks”. In: *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*. 2015.

[Els12] L. D. Elsgolc. *Calculus of variations*. Courier Corporation, 2012.

[Erh+09] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. *Visualizing higher-layer features of a deep network*. Tech. rep. 3. University of Montreal, 2009, p. 1.

[Evt+17] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song. “Robust Physical-World Attacks on Machine Learning Models”. In: *arXiv.org* 1707.08945 (2017).

[Far03] G. Farneback. “Two-Frame Motion Estimation Based on Polynomial Expansion”. In: *Scandinavian Conference on Image Analysis (SCIA)*. 2003.

[FH12] P. F. Felzenszwalb and D. P. Huttenlocher. “Distance Transforms of Sampled Functions”. In: *Theory of Computing* 8.1 (2012), pp. 415–428.

[FJ90] D. J. Fleet and A. D. Jepson. “Computation of component image velocity from local phase information”. In: *International Journal of Computer Vision (IJCV)* 5.1 (1990), pp. 77–104.

[FP10] Y. Furukawa and J. Ponce. “Accurate, Dense, and Robust Multi-View Stereopsis”. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)* 32.8 (2010), pp. 1362–1376.

[FPZ16] C. Feichtenhofer, A. Pinz, and A. Zisserman. “Convolutional Two-Stream Network Fusion for Video Action Recognition”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 1933–1941.

[Gar+16] R. Garg, B. G. V. Kumar, G. Carneiro, and I. D. Reid. “Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue”. In: *Proc. of the European Conf. on Computer Vision (ECCV)*. 2016.

[GBC16] I. J. Goodfellow, Y. Bengio, and A. C. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016. ISBN: 978-0-262-03561-3.

[Gei+13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. “Vision meets Robotics: The KITTI Dataset”. In: *International Journal of Robotics Research (IJRR)* 32.11 (2013), pp. 1231–1237.

[Gei09] A. Geiger. “Monocular road mosaicing for urban environments”. In: *Proc. IEEE Intelligent Vehicles Symposium (IV)*. 2009.

[GG15] F. Güney and A. Geiger. “Displets: Resolving Stereo Ambiguities using Object Knowledge”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[GG16] F. Güney and A. Geiger. “Deep Discrete Flow”. In: *Proc. of the Asian Conf. on Computer Vision (ACCV)*. 2016.

[GH02] T. Gautama and M. M. V. Hulle. “A phase-based approach to the estimation of the optical flow field using spatial filtering”. In: *Neural Networks* 13.5 (2002), pp. 1127–1136.

[Gib50] J. J. Gibson. *The perception of the visual world.* Houghton Mifflin, 1950.

[Gib58] J. J. Gibson. “Visually controlled locomotion and visual orientation in animals.” In: *British Journal of Psychology* 49 3 (1958), pp. 182–94.

[Gib66] J. J. Gibson. *The senses considered as perceptual systems.* Houghton Mifflin, 1966.

[GLU12] A. Geiger, P. Lenz, and R. Urtasun. “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[GMB17] C. Godard, O. Mac Aodha, and G. J. Brostow. “Unsupervised Monocular Depth Estimation with Left-Right Consistency”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[GPS89] D. M. Greig, B. T. Porteous, and A. H. Seheult. “Exact maximum a posteriori estimation for binary images”. In: *Journal of the Royal Statistical Society (JRSS)* 51.2 (1989), pp. 271–279.

[GR18] J. Gast and S. Roth. “Lightweight Probabilistic Deep Networks”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 3369–3378. DOI: `10.1109/CVPR.2018.00355`.

[GSS14] I. J. Goodfellow, J. Shlens, and C. Szegedy. “Explaining and harnessing adversarial examples”. In: *arXiv.org* (2014).

[He+17] Y. He, W. Chiu, M. Keuper, and M. Fritz. “STD2P: RGBD Semantic Segmentation Using Spatio-Temporal Data-Driven Pooling”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 7158–7167.

[Hee88] D. J. Heeger. “Optical flow using spatiotemporal filters”. In: *International Journal of Computer Vision (IJCV)* 1.4 (1988), pp. 279–302.

[HJ92] D. J. Heeger and A. D. Jepson. “Subspace methods for recovering rigid motion I: Algorithm and implementation”. In: *International Journal of Computer Vision (IJCV)* 7.2 (1992), pp. 95–117.

[HR19] J. Hur and S. Roth. “Iterative Residual Refinement for Joint Optical Flow and Occlusion Estimation”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 5754–5763.

[HS81] B. K. P. Horn and B. G. Schunck. “Determining Optical Flow”. In: *Artificial Intelligence (AI)* 17.1-3 (1981), pp. 185–203.

[HSS12] G. Hinton, N. Srivastava, and K. Swersky. “Neural networks for machine learning lecture 6a overview of mini-batch gradient descent”. In: *University of Toronto, Technical Report* (2012).

[Ilg+17] 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 Conf. on Computer Vision and Pattern Recognition (CVPR)* (2017).

[Ilg+18] E. Ilg, Ö. Çiçek, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Brox. “Uncertainty Estimates and Multi-hypotheses Networks for Optical Flow”. In: *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*. 2018, pp. 677–693.

[IM09] A. Ihler and D. McAllester. “Particle Belief Propagation”. In: *Conference on Artificial Intelligence and Statistics (AISTATS)*. 2009.

[Jad+15] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. “Spatial Transformer Networks”. In: *Advances in Neural Information Processing Systems (NIPS)*. 2015.

[Jan+17a] J. Janai, F. Güney, A. Behl, and A. Geiger. “Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art”. In: *arXiv.org* 1704.05519 (2017).

[Jan+17b] 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 Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[Jan+18] J. Janai, F. Güney, A. Ranjan, M. Black, and A. Geiger. “Unsupervised Learning of Multi-Frame Optical Flow with Occlusions”. In: *Proc. of the European Conf. on Computer Vision (ECCV)*. 2018.

[KB15] D. P. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization”. In: *Proc. of the International Conf. on Learning Representations (ICLR)*. 2015.

[Kim+14] H. Kim, A. Handa, R. Benosman, S. Ieng, and A. J. Davison. “Simultaneous Mosaicing and Tracking with an Event Camera”. In: *Proc. of the British Machine Vision Conf. (BMVC)*. 2014.

[KLA99] D. Koller, U. Lerner, and D. Anguelov. “A General Algorithm for Approximate Inference and Its Application to Hybrid Bayes Nets”. In: *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*. 1999, pp. 324–333.

[KMG08] C. Kondermann, R. Mester, and C. S. Garbe. “A Statistical Confidence Measure for Optical Flows”. In: *Proc. of the European Conf. on Computer Vision (ECCV)*. 2008.

[KN11] J. Kybic and C. Nieuwenhuis. “Bootstrap optical flow confidence and uncertainty measure”. In: *Computer Vision and Image Understanding (CVIU)* 115.10 (2011), pp. 1449–1462.

[Kol06] V. Kolmogorov. “Convergent Tree-Reweighted Message Passing for Energy Minimization”. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)* 28.10 (2006), pp. 1568–1583.

[Kon+07] C. Kondermann, D. Kondermann, B. Jähne, and C. S. Garbe. “An Adaptive Confidence Measure for Optical Flows Based on Linear Subspace Projections”. In: *Proc. of the German Conference on Pattern Recognition (GCPR)*. 2007, pp. 132–141.

[Kon+16] D. Kondermann, R. Nair, K. Honauer, K. Krispin, J. Andrulis, A. Brock, B. Gussefeld, M. Rahimimoghaddam, S. Hofmann, C. Brenner, and B. Jahne. “The HCI Benchmark Suite: Stereo and Flow Ground Truth With Uncertainties for Urban Autonomous Driving”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2016.

[Kro+16] T. Kroeger, R. Timofte, D. Dai, and L. V. Gool. “Fast Optical Flow Using Dense Inverse Search”. In: *Proc. of the European Conf. on Computer Vision (ECCV)*. 2016, pp. 471–488.

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems (NIPS)*. 2012.

[KT14] R. Kennedy and C. J. Taylor. “Optical Flow with Geometric Occlusion Estimation and Fusion of Multiple Frames”. In: *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*. 2014.

[KVK16] A. Kundu, V. Vineet, and V. Koltun. “Feature Space Optimization for Semantic Video Segmentation”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[KZ04] V. Kolmogorov and R. Zabih. “What Energy Functions Can Be Minimized via Graph Cuts?” In: *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)* 26.2 (2004), pp. 147–159.

[LAG04] S. Lim, J. G. Apostolopoulos, and A. E. Gamal. “Benefits of temporal oversampling in optical flow estimation”. In: *Proc. IEEE International Conf. on Image Processing (ICIP)*. 2004.

[LAG05] S. Lim, J. G. Apostolopoulos, and A. E. Gamal. “Optical flow estimation using temporally oversampled video”. In: *IEEE Trans. on Image Processing (TIP)* 14.8 (2005), pp. 1074–1087.

[Lee76] D. N. Lee. “A Theory of Visual Control of Braking Based on Information about Time-to-Collision”. In: *Perception* 5.4 (1976), pp. 437–459.

[LG01] S. Lim and A. E. Gamal. “Optical flow estimation using high frame rate sequences”. In: *Proc. IEEE International Conf. on Image Processing (ICIP)*. 2001.

[Lin+14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. “Microsoft COCO: Common Objects in Context”. In: *Proc. of the European Conf. on Computer Vision (ECCV)*. 2014.

[Liu+19] P. Liu, M. R. Lyu, I. King, and J. Xu. “SelFlow: Self-Supervised Learning of Optical Flow”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4571–4580.

[Liu+20] P. Liu, J. Janai, M. Pollefeys, T. Sattler, and A. Geiger. “Self-Supervised Motion Deblurring”. In: *IEEE Robotics and Automation Letters (RA-L)* (2020).

[LK81] B. D. Lucas and T. Kanade. “An Iterative Image Registration Technique with an Application to Stereo Vision”. In: *Proc. of the International Joint Conf. on Artificial Intelligence (IJCAI)*. 1981.

[Lon+16] G. Long, L. Kneip, J. M. Alvarez, H. Li, X. Zhang, and Q. Yu. “Learning Image Matching by Simply Watching Video”. In: *Proc. of the European Conf. on Computer Vision (ECCV)*. 2016.

[LV98] S. Lai and B. C. Vemuri. “Reliable and Efficient Computation of Optical Flow”. In: *International Journal of Computer Vision (IJCV)* 29.2 (1998), pp. 87–105.

[LZS13] M. Leordeanu, A. Zanfir, and C. Sminchisescu. “Locally affine sparse-to-dense matching for motion and occlusion estimation”. In: *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*. 2013.

[Ma+17] L. Ma, J. Stückler, C. Kerl, and D. Cremers. “Multi-view deep learning for consistent semantic mapping with RGB-D cameras”. In: *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*. 2017, pp. 598–605.

[Mac+13] O. Mac Aodha, A. Humayun, M. Pollefeys, and G. J. Brostow. “Learning a Confidence Measure for Optical Flow”. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)* 35.5 (2013), pp. 1107–1120.

[May+16] N. Mayer, E. Ilg, P. Haeusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. “A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[May+18] N. Mayer, E. Ilg, P. Fischer, C. Hazirbas, D. Cremers, A. Dosovitskiy, and T. Brox. “What Makes Good Synthetic Training Data for Learning Disparity and Optical Flow Estimation?” In: *International Journal of Computer Vision (IJCV)* (2018).

[MB18] D. Maurer and A. Bruhn. “ProFlow: Learning to Predict Optical Flow”. In: *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*. 2018, p. 86.

[MG15] M. Menze and A. Geiger. “Object Scene Flow for Autonomous Vehicles”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[MGS15] E. Mueggler, G. Gallego, and D. Scaramuzza. “Continuous-Time Trajectory Estimation for Event-based Vision Sensors”. In: *Proc. Robotics: Science and Systems (RSS)*. 2015.

[MHG15] M. Menze, C. Heipke, and A. Geiger. “Discrete Optimization for Optical Flow”. In: *Proc. of the German Conference on Pattern Recognition (GCPR)*. 2015.

[MHR18] S. Meister, J. Hur, and S. Roth. “UnFlow: Unsupervised Learning of Optical Flow with a Bidirectional Census Loss”. In: *Proc. of the Conf. on Artificial Intelligence (AAAI)*. 2018.

[Mue+15] E. Mueggler, C. Forster, N. Baumli, G. Gallego, and D. Scaramuzza. “Lifetime estimation of events from Dynamic Vision Sensors”. In: *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*. 2015.

[Mül+11] T. Müller, C. Rabe, J. Rannacher, U. Franke, and R. Mester. “Illumination-Robust Dense Optical Flow Using Census Signatures”. In: *Proc. of the DAGM Symposium on Pattern Recognition (DAGM)*. 2011, pp. 236–245.

[NYC15] A. M. Nguyen, J. Yosinski, and J. Clune. “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[ODO16] A. Odena, V. Dumoulin, and C. Olah. “Deconvolution and Checkerboard Artifacts”. In: *Distill* (2016). URL: `http://distill.pub/2016/deconv-checkerboard/`.

[Pan12] G. Panin. “Mutual Information for Multi-modal, Discontinuity-Preserving Image Registration”. In: *Proc. of the International Symposium on Advances in Visual Computing (ISVC)*. 2012, pp. 70–81.

[PHC16] V. Pătrăucean, A. Handa, and R. Cipolla. “Spatio-temporal video autoencoder with differentiable memory”. In: *Proc. of the International Conf. on Learning Representations (ICLR)*. 2016.

[Ran+19a] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black. "Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 12240–12249.

[Ran+19b] A. Ranjan, J. Janai, A. Geiger, and M. Black. "Attacking Optical Flow". In: *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*. 2019.

[RB16] A. Ranjan and M. J. Black. "Optical Flow Estimation using a Spatial Pyramid Network". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* 1611.00850 (2016).

[RBP14] R. Ranftl, K. Bredies, and T. Pock. "Non-local Total Generalized Variation for Optical Flow Estimation". In: *Proc. of the European Conf. on Computer Vision (ECCV)*. 2014.

[RDR13] J. Ralli, J. Díaz, and E. Ros. "Spatial and temporal constraints in variational correspondence methods". In: *Machine Vision and Applications (MVA)* 24.2 (2013), pp. 275–287.

[Ren+17] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha. "Unsupervised Deep Learning for Optical Flow Estimation". In: *Proc. of the Conf. on Artificial Intelligence (AAAI)*. 2017.

[Rev+15] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. "EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[RHK17] S. R. Richter, Z. Hayder, and V. Koltun. "Playing for Benchmarks". In: *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*. 2017.

[Ric+16] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. "Playing for Data: Ground Truth from Computer Games". In: *Proc. of the European Conf. on Computer Vision (ECCV)*. 2016.

[Rus+15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252.

[SAH91] E. P. Simoncelli, E. H. Adelson, and D. J. Heeger. "Probability distributions of optical flow". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 1991.

[Sch+18] R. Schuster, C. Bailer, O. Wasenmüller, and D. Stricker. "FlowFields++: Accurate Optical Flow Correspondences Meet Robust Interpolation". In: *Proc. IEEE International Conf. on Image Processing (ICIP)*. 2018, pp. 1463–1467.

[SCH15] M. Savva, A. X. Chang, and P. Hanrahan. "Semantically-Enriched 3D Models for Common-sense Knowledge". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops* (2015).

[Sev+16] L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black. “Optical Flow with Semantic Segmentation and Localized Layers”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[Sha+16] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter. “Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition”. In: *Proc. of the ACM SIGSAC Conference on Computer and Communications Security*. 2016, pp. 1528–1540.

[SPC09] F. Steinbrücker, T. Pock, and D. Cremers. “Advanced Data Terms for Variational Optic Flow Estimation”. In: *Vision, Modeling and Visualization (VMV)*. 2009, pp. 155–164.

[SRB10] D. Sun, S. Roth, and M. J. Black. “Secrets of optical flow estimation and their principles”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2010.

[SRB14] D. Sun, S. Roth, and M. J. Black. “A Quantitative Analysis of Current Practices in Optical Flow Estimation and the Principles Behind Them”. In: *International Journal of Computer Vision (IJCV)* 106.2 (2014), pp. 115–137.

[SS07] A. Salgado and J. Sánchez. “Temporal Constraints in Large Optical Flow”. In: *Proc. of the International Conf. on Computer Aided Systems Theory (EUROCAST)*. 2007.

[SSB10] D. Sun, E. B. Sudderth, and M. J. Black. “Layered image motion with explicit occlusions, temporal consistency, and depth ordering”. In: *Advances in Neural Information Processing Systems (NIPS)*. 2010.

[SSB12] D. Sun, E. B. Sudderth, and M. J. Black. “Layered segmentation and optical flow estimation over time”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[ST08] P. Sand and S. Teller. “Particle Video: Long-Range Motion Estimation Using Point Trajectories”. In: *International Journal of Computer Vision (IJCV)* 80.1 (2008), p. 72.

[Sun+13] D. Sun, J. Wulff, E. Sudderth, H. Pfister, and M. Black. “A fully-connected layered model of foreground and background flow”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[Sun+18a] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. “Models Matter, So Does Training: An Empirical Study of CNNs for Optical Flow Estimation”. In: *arXiv.org* (2018).

[Sun+18b] 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 Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[SVB13] M. Stoll, S. Volz, and A. Bruhn. “Joint trilateral filtering for multiframe optical flow”. In: *Proc. IEEE International Conf. on Image Processing (ICIP)*. 2013.

[SZ14] K. Simonyan and A. Zisserman. “Two-Stream Convolutional Networks for Action Recognition in Videos”. In: *Advances in Neural Information Processing Systems (NIPS)*. 2014, pp. 568–576.

[Sze+13] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. “Intriguing properties of neural networks”. In: *arXiv.org* (2013).

[Sze11] R. Szeliski. *Computer Vision - Algorithms and Applications*. Texts in Computer Science. Springer, 2011.

[TG15] R. Timofte and L. V. Gool. “Sparse Flow: Sparse Matching for Small to Large Displacement Optical Flow”. In: *Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2015.

[TM09] H. Trinh and D. McAllester. “Unsupervised Learning of Stereo Vision with Monocular Cues”. In: *Proc. of the British Machine Vision Conf. (BMVC)*. 2009.

[Ura+88] S. Uras, F. Girosi, A. Verri, and V. Torre. “A computational approach to motion perception”. In: *Biological Cybernetics* 60.2 (1988), pp. 79–87.

[VBW08] L. Valgaerts, A. Bruhn, and J. Weickert. “A Variational Model for the Joint Recovery of the Fundamental Matrix and the Optical Flow”. In: *Proc. of the DAGM Symposium on Pattern Recognition (DAGM)*. 2008, pp. 314–324.

[Vij+17] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. “SfM-Net: Learning of Structure and Motion from Video”. In: *arXiv.org* 1704.07804 (2017).

[Vol+11] S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer. “Modeling temporal coherence for optical flow”. In: *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*. 2011.

[Wan+18a] S. Wang, Y. Zhou, J. Yan, and Z. Deng. “Fully Motion-Aware Network for Video Object Detection”. In: *Proc. of the European Conf. on Computer Vision (ECCV)*. 2018.

[Wan+18b] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu. “Occlusion Aware Unsupervised Learning of Optical Flow”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[WB15] J. Wulff and M. J. Black. “Efficient Sparse-to-Dense Optical Flow Estimation using a Learned Basis and Layers”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[Wei+13] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. “DeepFlow: Large Displacement Optical Flow with Deep Matching”. In: *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*. 2013.

[WFW08] C. M. Wang, K. C. Fan, and C. T. Wang. “Estimating optical flow by integrating multi-frame information”. In: *Journal of Information Science and Engineering (JISE)* (2008).

[WKR17] A. S. Wannenwetsch, M. Keuper, and S. Roth. “ProbFlow: Joint Optical Flow and Uncertainty Estimation”. In: *Proc. of the IEEE International Conf. on Computer Vision (ICCV).* 2017, pp. 1182–1191. DOI: `10.1109/ICCV.2017.133`.

[WLF14] D. Wei, C. Liu, and W. Freeman. “A Data-driven Regularization Model for Stereo and Flow”. In: *Proc. of the International Conf. on 3D Vision (3DV).* 2014.

[Woj+17] Z. Wojna, J. R. R. Uijlings, S. Guadarrama, N. Silberman, L. Chen, A. Fathi, and V. Ferrari. “The Devil is in the Decoder: Classification, Regression and GANs”. In: *Proc. of the British Machine Vision Conf. (BMVC).* 2017.

[WS01] J. Weickert and C. Schnörr. “Variational optic flow computation with a spatio-temporal smoothness constraint”. In: *Journal of Mathematical Imaging and Vision (JMIV)* 14.3 (2001), pp. 245–255.

[WSB17] J. Wulff, L. Sevilla-Lara, and M. J. Black. “Optical Flow in Mostly Rigid Scenes”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR).* 2017, pp. 6911–6920. DOI: `10.1109/CVPR.2017.731`. URL: `https://doi.org/10.1109/CVPR.2017.731`.

[Wu13] C. Wu. “Towards Linear-time Incremental Structure from Motion”. In: *Proc. of the International Conf. on 3D Vision (3DV).* 2013.

[XGF16] J. Xie, R. B. Girshick, and A. Farhadi. “Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks”. In: *Proc. of the European Conf. on Computer Vision (ECCV).* 2016.

[XJM12] L. Xu, J. Jia, and Y. Matsushita. “Motion Detail Preserving Optical Flow Estimation”. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)* 34.9 (2012), pp. 1744–1757.

[YHD16] 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. of the European Conf. on Computer Vision (ECCV).* 2016.

[YK16] F. Yu and V. Koltun. “Multi-Scale Context Aggregation by Dilated Convolutions”. In: *Proc. of the International Conf. on Learning Representations (ICLR).* 2016.

[YL15] J. Yang and H. Li. “Dense, Accurate Optical Flow Estimation with Piecewise Parametric Model”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR).* 2015.

[YMU13] K. Yamaguchi, D. McAllester, and R. Urtasun. “Robust Monocular Epipolar Flow Estimation”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR).* 2013.

[YMU14] K. Yamaguchi, D. McAllester, and R. Urtasun. “Efficient joint segmentation, occlusion labeling, stereo and flow estimation”. In: *Proc. of the European Conf. on Computer Vision (ECCV).* 2014.

[Yos+15] J. Yosinski, J. Clune, A. M. Nguyen, T. J. Fuchs, and H. Lipson. “Understanding Neural Networks Through Deep Visualization”. In: *arXiv.org* abs/1506.06579 (2015).

[ZBW11] H. Zimmer, A. Bruhn, and J. Weickert. “Optic flow in harmony”. In: *International Journal of Computer Vision (IJCV)* 93.3 (2011), pp. 368–388.

[Zho+17] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. “Unsupervised Learning of Depth and Ego-Motion from Video”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[Zhu+17a] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei. “Flow-Guided Feature Aggregation for Video Object Detection”. In: *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*. 2017.

[Zhu+17b] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei. “Deep Feature Flow for Video Recognition”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[Zhu+18] X. Zhu, J. Dai, L. Yuan, and Y. Wei. “Towards High Performance Video Object Detection”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[ŽL16] J. Žbontar and Y. LeCun. “Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches”. In: *Journal of Machine Learning Research (JMLR)* 17.65 (2016), pp. 1–32.

[ZPB07] C. Zach, T. Pock, and H. Bischof. “A Duality Based Approach for Realtime TV-L1 Optical Flow”. In: *Proc. of the DAGM Symposium on Pattern Recognition (DAGM)*. 2007, pp. 214–223.