TECHNISCHE
UNIVERSITÄT
DARMSTADT

JOINT MOTION, SEMANTIC SEGMENTATION, OCCLUSION, AND DEPTH ESTIMATION

A dissertation submitted to
TECHNISCHE UNIVERSITÄT DARMSTADT
Fachbereich Informatik

in fulfillment of the requirements for the degree of
Doktor-Ingenieur (Dr.-Ing.)

presented by

JUNHWA HUR
M.Sc.

born in Seoul, Korea

Examiner:	Prof. Stefan Roth, Ph.D.
Co-examiner:	Prof. Deva Ramanan, Ph.D.
Date of Submission:	5th April 2022
Date of Defense:	18th May 2022

Darmstadt, 2022

Joint Motion, Semantic Segmentation, Occlusion, and Depth Estimation

Submitted doctoral thesis by Junhwa Hur

Examiner: Prof. Stefan Roth, Ph.D.
Co-examiner: Prof. Deva Ramanan, Ph.D.

Date of submission: 5th April 2022
Date of thesis defense: 18th May 2022

Darmstadt, Technische Universität Darmstadt
Jahr der Veröffentlichung der Dissertation auf TUprints: 2022

Bitte zitieren Sie dieses Dokument als:
URN: urn:nbn:de:tuda-tuprints-216242
URL: http://tuprints.ulb.tu-darmstadt.de/21624

Dieses Dokument wird bereitgestellt von tuprints,
E-Publishing-Service der TU Darmstadt
http://tuprints.ulb.tu-darmstadt.de
tuprints@ulb.tu-darmstadt.de

ABSTRACT

Visual scene understanding is one of the most important components of autonomous navigation. It includes multiple computer vision tasks such as recognizing objects, perceiving their 3D structure, and analyzing their motion, all of which have gone through remarkable progress over the recent years. However, most of the earlier studies have explored these components individually, and thus potential benefits from exploiting the relationship between them have been overlooked. In this dissertation, we explore what kind of relationship the tasks can present, along with the potential benefits that could be discovered from jointly formulating multiple tasks. The joint formulation allows each task to exploit the other task as an additional input cue and eventually improves the accuracy of the joint tasks.

We first present the joint estimation of semantic segmentation and optical flow. Though not directly related, the tasks provide an important cue to each other in the temporal domain. Semantic information can provide information on plausible physical motion of its associated pixels, and accurate pixel-level temporal correspondences enhance the temporal consistency of semantic segmentation. We demonstrate that the joint formulation improves the accuracy of both tasks.

Second, we investigate the mutual relationship between optical flow and occlusion estimation. Unlike most previous methods considering occlusions as outliers, we highlight the importance of jointly reasoning the two tasks in the optimization. Specifically through utilizing forward-backward consistency and occlusion-disocclusion symmetry in the energy, we demonstrate that the joint formulation brings substantial performance benefits for both tasks on standard benchmarks.

We further demonstrate that optical flow and occlusion can exploit their mutual relationship in Convolutional Neural Network as well. We propose to iteratively and residually refine the estimates using a single weight-shared network, which substantially improves the accuracy without adding network parameters or even reducing them depending on the backbone networks.

Next, we propose a joint depth and 3D scene flow estimation from only two temporally consecutive monocular images. We solve this ill-posed problem by taking an inverse problem view. We design a single Convolutional Neural Network that simultaneously estimates depth and 3D motion from a classical optical flow cost volume. With self-supervised learning, we leverage unlabeled data for training, without concerns about the shortage of 3D annotation for direct supervision.

Finally, we conclude by summarizing the contributions and discussing future perspectives that can resolve current challenges our approaches have.

ZUSAMMENFASSUNG

Das visuelle Szenenverständnis ist eine der wichtigsten Komponenten der autonomen Navigation. Es umfasst mehrere Computer-Vision-Aufgaben wie das Erkennen von Objekten, das Wahrnehmen ihrer 3D-Struktur und die Analyse ihrer Bewegung, die in den letzten Jahren alle bemerkenswerte Fortschritte gemacht haben. In den meisten früheren Studien wurden diese Aufgaben jedoch einzeln untersucht, und daher wurden potenzielle Vorteile aus der Nutzung der Beziehung zwischen ihnen übersehen. In dieser Dissertation untersuchen wir, welche Aufgaben miteinander in Verbindung stehen und welche potenziellen Vorteile sich aus der gemeinsamen Formulierung mehrerer Aufgaben ergeben könnten. Die gemeinsame Formulierung ermöglicht es jeder Aufgabe, die andere Aufgabe als zusätzlichen Eingabehinweis zu nutzen und verbessert schließlich die Genauigkeit der gemeinsamen Aufgaben.

Wir präsentieren zunächst die gemeinsame Schätzung von semantischer Segmentierung und optischem Fluss. Obwohl diese Probleme nicht direkt miteinander verbunden sind, geben die Aufgaben im zeitlichen Bereich einen wichtigen Hinweis aufeinander. Semantische Informationen können ihren zugeordneten Pixeln Informationen über eine plausible physikalische Bewegung bieten, und genaue zeitliche Korrespondenzen auf Pixelebene verbessern die zeitliche Konsistenz der semantischen Segmentierung. Wir zeigen, dass die gemeinsame Formulierung die Genauigkeit beider Aufgaben verbessert.

Zweitens untersuchen wir die gegenseitige Beziehung zwischen optischem Fluss und Okklusionsschätzung. Im Gegensatz zu den meisten früheren Methoden, die Okklusionen als Ausreißer betrachten, betonen wir die Wichtigkeit der gemeinsamen Schätzung der beiden Aufgaben bei der Optimierung. Insbesondere durch die Verwendung von Vorwärts-Rückwärts-Konsistenz und Okklusions-Disokklusions-Symmetrie in der Energie zeigen wir, dass die gemeinsame Formulierung erhebliche Leistungsvorteile für beide Aufgaben bei Standard-Benchmarks bringt.

Wir zeigen weiter, dass sich optischer Fluss und Okklusion auch in Convolutional Neural Networks gegenseitig ausnutzen können. Wir schlagen vor, die Schätzungen iterativ und schrittweise zu verfeinern, indem ein Netzwerk mit gemeinsamen Gewichtsparameter verwendet wird, was die Genauigkeit erheblich verbessert, ohne Netzwerkparameter hinzuzufügen oder diese sogar zu reduzieren, je nach Netzwerkarchitektur.

Dann schlagen wir eine gemeinsame Tiefen- und 3D-Szenenflussschätzung aus nur zwei zeitlich aufeinanderfolgenden monokularen Bildern vor. Wir lösen dieses unterbestimmte Problem durch eine inverse Problemsicht. Wir entwerfen ein einzelnes Convolutional Neural Network, das gleichzeitig Tiefe und 3D-Bewegung aus einem klassischen optischen Flusskostenvolumen schätzt. Beim selbstüberwachten Lernen nutzen wir Daten ohne Annotationen für das Training, ohne Bedenken hinsichtlich des Fehlens von 3D-Annotationen für die direkte Überwachung.

Abschließend fassen wir die Beiträge zusammen und diskutieren Zukunftsperspektiven, die aktuelle Herausforderungen unserer Methoden lösen können.

PUBLICATIONS

Junhwa Hur and Stefan Roth
Self-Supervised Multi-Frame Monocular Scene Flow. In *Proceedings of the IEEE/CVF Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual, June 2021, pp. 2684–2694.

Junhwa Hur and Stefan Roth
Self-Supervised Monocular Scene Flow Estimation. In *Proceedings of the IEEE/CVF Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual, June 2020, pp. 7396–7405.

Junhwa Hur and Stefan Roth
Optical Flow Estimation in the Deep Learning Age. In *Modelling Human Motion*, Ed. by Nicoletta Noceti, Alessandra Sciutti, and Francesco Rea, Springer, 2020, pp. 119–140.

Junhwa Hur and Stefan Roth
Iterative Residual Refinement for Joint Optical Flow and Occlusion Estimation. In *Proceedings of the IEEE/CVF Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, California, June 2019, pp. 5754–5763.

Simon Meister, Junhwa Hur, and Stefan Roth
UnFlow: Unsupervised Learning of Optical Flow with a Bidirectional Census Loss. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, New Orleans, Louisiana, February 2018, pp. 7251–7259.

Junhwa Hur and Stefan Roth
MirrorFlow: Exploiting Symmetries in Joint Optical Flow and Occlusion Estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017, pp. 312–321.

Junhwa Hur and Stefan Roth
Joint Optical Flow and Temporally Consistent Semantic Segmentation. In *4th Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving, jointly with ECCV*, Amsterdam, The Netherlands, October 2016, pp. 163–177.

ACKNOWLEDGMENTS

First and foremost, I am truly thankful for my advisor Stefan Roth for his time, continuous support, and advice during my Ph.D. study. I appreciate his countless help on research: the freedom of choosing topics, guidance on research directions, helpful and critical comments that improve work, practical tips for debugging algorithms, and writing skills as well. Not only that, he taught me a lot about managing to work efficiently, such as time management, context switching, and organizing tasks upon their priorities.

I also would like to deeply thank my committee members, Prof. Deva Ramanan, Ph.D., Prof. Dr. Kristian Kersting, Prof. Georgia Chalvatzaki, Ph.D., and Prof. Dr. Zsolt István, for their time, insightful comments, valuable feedbacks, and suggestions.

I am very thankful for my colleagues, Stephan Richter, Tobias Plötz, Jochen Gast, Anne Wannenwetsch, Faraz Saeedan, Nikita Araslanov, Shweta Mahajan, and Xiang Chen, for sharing helpful discussion on research and making the journey to the Ph.D. more enjoyable. It was nice to meet new colleagues, Jannik Schmitt, Jan-Martin Steitz, and Simone Schaub-Meyer, although we did not have much time to meet in person from early 2020 due to the home office regulations. I miss those lunch time, coffee and espresso time, and muffin time that relieve the stress from research, teaching, and especially the paper deadlines. I am grateful for Marius Cordts and Jochen Gast for sharing a pre-trained model and training modules for research and publications. Without their helps, meeting deadlines for paper submissions would have been more difficult and delayed. I very much thank Simon Meister for the co-work on a publication, including the in-depth discussions on the work and his motivation that I learned a lot from him. I am grateful for Apratim Bhattacharyya, Sungho Jeon, and Jinseok Nam for sharing their valuable experience and advice on Ph.D. journey, which comforted me and gave me courage. Also, I truly appreciate Nicole Schätzle and Horst Fey for the enormous help on administration issues.

Last but not least, I deeply appreciate the support and loves from my parents and sister throughout all of these years.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AEPE	Average End-Point Error
ANN	Approximate Nearest Neighbor
Bi	Bi-Directional
CAD	Computer-Aided Design
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CRF	Conditional Random Field
DNN	Deep Neural Network
DoF	Degrees of Freedom
EPE	End-Point Error
FCN	Fully Convolutional Network
GPU	Graphics Processing Unit
HoG	Histogram of oriented Gradients (Dalal and Triggs, 2005)
GRU	Gated Recurrent Unit
IoU	Intersection over Union
IRR	Iterative Residual Refinement
KITTI	Karlsruhe Institute of Technology (KIT) and Toyota Technological Institute at Chicago (TTI-C)
LiDAR	Light Detection and Ranging
LK	Lucas Kanade (Lucas and Kanade, 1981)
MAP	Maximum A Posteriori
MPI	Max Planck Institute
MRF	Markov Random Field
NCC	Normalized Cross Correlation
Occ	Occlusion
OoB	Out-of-Bound
PMBP	PatchMatch Belief Propagation (Besse et al., 2013)

QPBO	Quadratic Pseudo-Boolean Optimization
RADAR	RAdio Detection And Ranging
RANSAC	RANdom SAmple Consensus (Fischler and Bolles, 1981)
RGB-D	Red, Green, Blue, and Depth
SIFT	Scale-Invariant Feature Transform (Lowe, 2004)
SGM	Semi-Global block Matching (Hirschmüller, 2008)
SfM	Structure from Motion

1

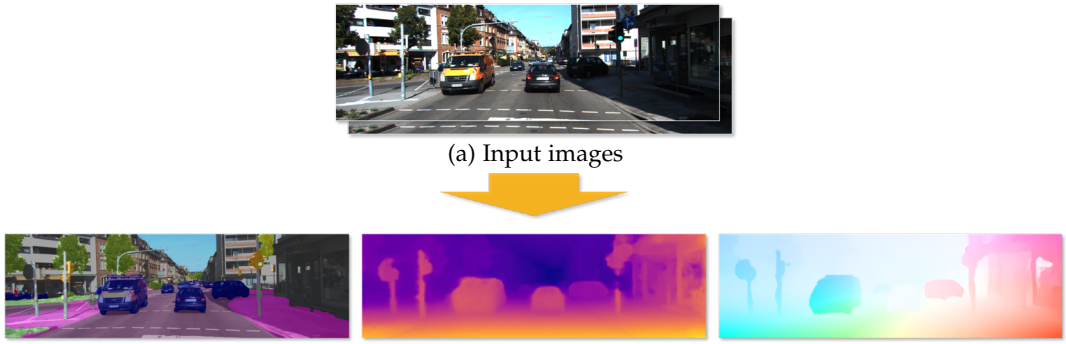
INTRODUCTION

CONTENTS

Along with growing needs and expectations for autonomous navigation, there have been continuing efforts on researching computer vision for 3D visual scene understanding. For safe and successful autonomous navigation, an intelligent system needs to holistically understand the scene, and corresponding computer vision research includes *(i)* surrounding environment recognition, *(ii)* 3D structure perception, and *(iii)* motion analysis. Over the recent years, public benchmark datasets have shown remarkable progress on solving these computer vision tasks. However, previous work has primarily treated these tasks individually and thus, potential benefits that can be exploited from their joint formulation remain unexplored. In this dissertation, we aim to demonstrate that a joint formulation can help each task leverage the other and resolve ambiguities that single task formulations cannot address efficiently. We first provide a brief introduction of major tasks for 3D visual scene understanding in Section 1.1. Then, we highlight the importance and motivation toward the joint estimation in Section 1.2. Finally, Section 1.3 summarizes the main contributions and gives brief outline of the dissertation.

1.1 PROBLEM STATEMENT

Humans primarily rely on their visual perception intelligence when navigating the 3D world. With electrical signals sent from the eyes and processed in the visual cortex, humans understand the surrounding environment and interact accordingly. Likewise for autonomous navigation systems, computer vision plays a crucial role in their



(a) Input images

(b) Semantic segmentation (c) Depth estimation (d) Motion estimation

Figure 1.1: For autonomous navigation, an intelligent system needs to holistically understand the scene by *(a)* recognizing objects, *(b)* perceiving their 3D structure, and *(c)* analyzing their motion as well.

visual perception intelligence. From input images captured by an on-board camera, a computer vision system processes the images and provides visual cues of the scene, which then can be utilized for a high-level decision-making process, such as path planning, potential hazard avoidance, or even future prediction.

Building a computer vision system for autonomous navigation remains challenging, coupled with a variety of computer vision problems. To achieve a comprehensive scene understanding, the system needs to solve a series of computer vision problems: it needs to recognize objects existing in the scene (*e. g.*, semantic segmentation, *cf*. Fig. 1.1b), perceive their 3D structure (*e. g.*, depth estimation, *cf*. Fig. 1.1c), and analyze the motion of each entity in the scene (*e. g.*, motion estimation, *cf*. Fig. 1.1d). Here, we briefly discuss each topic along with its corresponding challenges that lead toward the main motivation of this dissertation.

Sensor modality. To solve such tasks, different types of data could be utilized, such as monocular, stereo or RGB-D images, or 3D point data. For real-world applications, data with depth cues are generally favored because the given depth cues provide geometric information of the scene and thus can improve the accuracy of perception tasks. In this dissertation, however, we only consider two temporally consecutive monocular frames from a monocular camera as an input. Despite a potential disadvantage in accuracy, the monocular camera demonstrates several advantages comparing to other sensor modalities: it is more economical (*vs*. LiDAR), is applicable to both indoor and outdoor usage (*vs*. RGB-D), and doesn't require sensor calibration (*vs*. stereo camera). Thus, our approach can be practical but also applicable to more general application domains such as web videos or mobile devices with a monocular camera, unlike methods using other sensor modalities. We demonstrate our contributions on this simple yet very challenging setup, and consider extensions to a multi-frame setup or a usage of additional inputs (*e. g.* depth cues) as future work. Yet, we don't really overlook the advantage of other sensor modalities. In Chapter 6, we also demonstrate using stereo images for helpful learning signals at training time, while we use a monocular camera and keep its main benefits at test time.

1.1.1 *Motion estimation*

Motion information (*cf*. Fig. 1.1d) is one of the most important cues for visual scene understanding in the temporal domain. Knowing where objects that are present in the scene move, an intelligent system is able to analyze and even help predict their future trajectories in the scene. Depending on the spatial coordinate, the task can be defined as 2D optical flow estimation, that is motion in image coordinates, or 3D scene flow estimation in 3D world coordinates.

Optical flow. According to Horn and Schunck (1981), optical flow is the apparent motion of brightness patterns between two temporally consecutive images. As one of the most studied topics in computer vision, the extent of our understanding has remarkably progressed for the past few decades. Yet, ongoing release of public benchmark datasets (Baker et al., 2011; Butler et al., 2012; Menze et al., 2015b; 2018) has kept uncovering more challenges and limitations that existing methods encounter.

Such challenges mainly come from where matching ambiguity exists. Complex large motion, illumination changes, or textureless areas make it difficult to find the matched brightness pattern without any types of spatial regularization, such as local smoothness or a motion prior. Also, occlusion that occurs due to moving objects or camera ego-motion makes the problem more challenging because its correspondence doesn't exist. From the perspective of how humans perceive, the matching ambiguity can be effectively resolved by exploiting geometric or semantic cues (*e. g.*, rigid motion representation or motion estimation in an object instance level), yet those aspects remain relatively underexplored.

Scene flow. Scene flow estimation is the task of obtaining 3D structure and 3D motion of dynamic scenes (Vedula et al., 1999; 2005). It has been receiving increasing attention for real-world applications, such as autonomous driving, robotics, and virtual/augmented reality, where 3D information is critical.

Consequently, many scene flow approaches have been proposed recently, based on different types of input data such as stereo images (Huguet and Devernay, 2007; Schuster et al., 2018b; Vogel et al., 2013b; Wedel et al., 2011; Zhang and Kambhamettu, 2001), 3D point clouds (Gu et al., 2019; Liu et al., 2019d), or a sequence of RGB-D images (Hadfield and Bowden, 2011; Hornáček et al., 2014; Lv et al., 2018; Qiao et al., 2018; Quiroga et al., 2014; Thakur and Mukherjee, 2018). Regardless of the type, the same technical challenges persist as with optical flow estimation (*e. g.*, occlusion, large displacement, and matching ambiguity) as they share a similar objective, *i. e.*, a correspondence matching task.

1.1.2 *Depth estimation*

Depth cues (*cf*. Fig. 1.1c) provide not only a 3D measurement but also a geometrical view of the scene, benefiting many real-world computer vision tasks, such as 3D reconstruction (Izadi et al., 2011), 3D object detection (Song and Xiao, 2014), segmentation (Gupta et al., 2014), etc. Depending on the type of sensing device, obtaining

depth cues comes with only minimal processing cost (*e. g.*, LiDAR, RADAR, or RGB-D cameras) or requires to solve a correspondence problem between two stereo images.

On the one hand, using a monocular camera for depth estimation, which is our main interest, has received increasing attention, due to its economical sensor setup and applicability to where only monocular images are available. Despite of its ill-posed property such as depth and scale ambiguity, studies have introduced various types of approaches, such as *(i)* a direct supervision to learn scene prior knowledge (Eigen et al., 2014), *(ii)* relative depth estimation from either motion parallax (Rogers and Graham, 1979) or epipolar motion (Ranjan et al., 2019; Yin and Shi, 2018; Zhou et al., 2017), or *(iii)* self-supervision from stereo image pairs (Godard et al., 2017). Yet, the main challenges persist, including low accuracy (Zhou et al., 2017), poor generalization to unknown scenes (Laina et al., 2016), and inaccurate depth estimates on moving objects (Ranjan et al., 2019; Yin and Shi, 2018).

1.1.3 *Semantic segmentation*

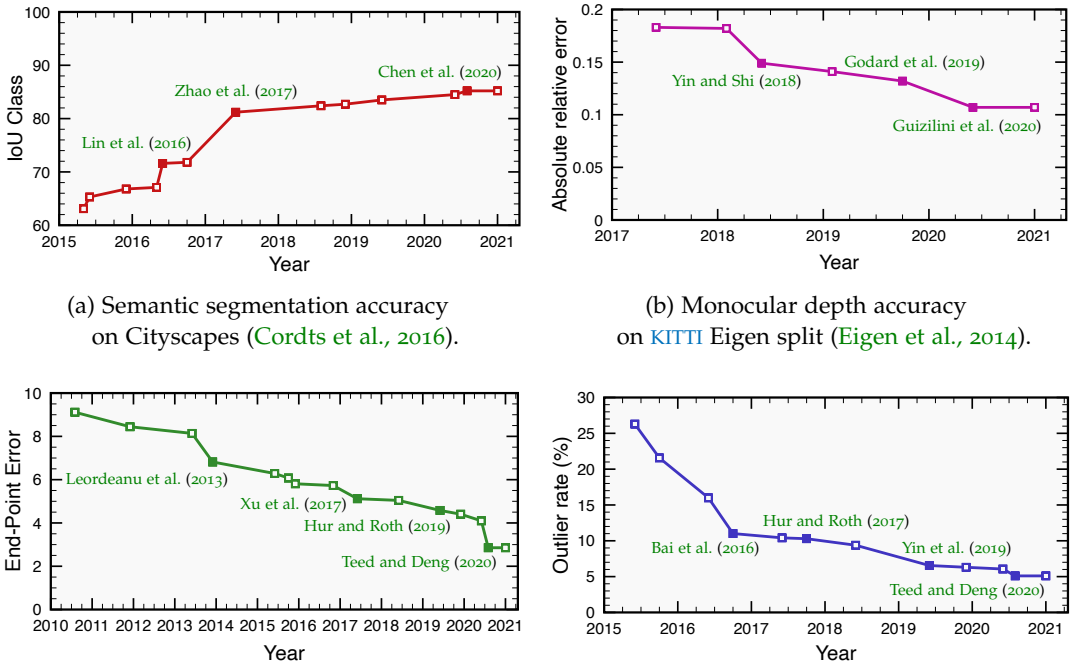
Semantic segmentation (*cf*. Fig. 1.1b) classifies each pixel of an image into one of the pre-defined object class labels. Such dense per-pixel object information enables a contextual scene understanding. In the context of autonomous driving, an autonomous car should recognize what objects are present around it to properly understand the circumstances and make a corresponding decision, for example, which way to drive or how to behave in a specific situation.

To correctly identify the semantic class of each pixel, one needs to robustly handle both photometric and geometric variation (*e. g.*, scale, view, shape) of objects belonging to the same class. Over the recent decade, semantic segmentation has gone through remarkable advances in technology and accuracy, attributable to evolving Convolutional Neural Networks (CNNs) and large-scale annotated datasets (Cordts et al., 2016; Neuhold et al., 2017) as well as synthetic datasets (Richter et al., 2017; 2016). Yet, challenges remain, such as *(i)* the dependence on annotated data, *(ii)* thus, a lack of generalization to different domains, and *(iii)* missing connections to other problem domains (*e. g.*, optical flow or depth estimation) for a higher level of scene understanding. All of these challenges warrant further investigation.

1.2 MOTIVATION TOWARD JOINT ESTIMATION

Active research and recent advances in deep learning have driven substantial progress on each task addressed above (*i. e.*, motion estimation, depth estimation, and semantic segmentation). Fig. 1.2 shows the state-of-the-art accuracy of each task in public benchmark datasets (Baker et al., 2011; Butler et al., 2012; Cordts et al., 2016; Eigen et al., 2014; Menze et al., 2015b; 2018) over the recent years. The accuracy improvement has been mainly achieved by:

- Novel CNN architectures and representations, such as pyramid-based architectures (Yin et al., 2019; Zhao et al., 2017), 3D convolutions (Guizilini et al., 2020), and 4D cost volumes (Teed and Deng, 2020; Xu et al., 2017).



(a) Semantic segmentation accuracy on Cityscapes (Cordts et al., 2016).

(b) Monocular depth accuracy on KITTI Eigen split (Eigen et al., 2014).

(c) Optical flow accuracy on Sintel (Butler et al., 2012)

(d) Optical flow accuracy on KITTI (Menze et al., 2015b)

Figure 1.2: **State-of-the-art accuracy on public benchmark datasets over recent years:** *(a)* Semantic segmentation accuracy on the Cityscapes dataset (Cordts et al., 2016), *(b)* monocular depth accuracy on the KITTI Eigen split (Eigen et al., 2014), and *(c)* optical flow accuracy on Sintel (Butler et al., 2012) and *(d)* on KITTI (Menze et al., 2015b).

- Better (proxy) loss functions (Godard et al., 2019; Jonschkowski et al., 2020) for (self-)supervised learning.
- Better training strategies, such as piecewise training (Lin et al., 2016), leveraging unlabeled data (Chen et al., 2020), augmentation schemes (Bar-Haim and Wolf, 2020), and learning rate schedules (Sun et al., 2020).

However, most of the existing methods have primarily focused on improving an individual task only; the mutual connections between the tasks remain underinvestigated. Though these tasks may not be closely related on a first glance, a joint formulation of the tasks can help them benefit each other and resolve ambiguities that cannot be solved with a single task formulation. For example, semantic segmentation can provide motion estimation with prior knowledge of physically plausible motion depending on object types. Temporal correspondence between two consecutive frames with a moving camera can resolve the depth ambiguity by exploiting motion parallax when estimating depth with a monocular camera. Also, depth or motion cues can improve semantic segmentation accuracy via encouraging spatially or temporally consistent estimation.

We, humans, do not always solve those perception tasks individually but rather try to understand a scene holistically from a mixture of low- and high-level visual

cues. This can resolve existing visual ambiguity by connecting different objects and surroundings (*cf*., Gestalt psychology (Ellis, 1938)). A performance improvement of a single task is important to achieve a technical advance; however more importantly, with a joint estimation approach, tasks can benefit from each other, which can ultimately help enrich a holistic scene understanding.

In this dissertation, we investigate the mutual dependencies between each task mentioned above and joint formulations that each task can leverage the other. In each chapter, we define each joint estimation case and propose the corresponding technical contributions. This joint formulation is case-specific, depending on the type of the relationship between tasks (*e. g*., highly to weakly related or being a subtask of another task). We explore which technical design of their mutual dependency can bring benefits in each case. We also demonstrate these benefits in both traditional energy-based models and recent CNN-based models.

Earlier studies (Bai et al., 2016; Bailer et al., 2015; Gadde et al., 2017; Nilsson and Sminchisescu, 2018; Sevilla-Lara et al., 2016; Yang and Ramanan, 2020) have demonstrated using the output from one task to supplement the other and focused on a main task only (more discussion and literature reviews follow in Section 2.3). Yet, these off-the-shelf methods are not accurate enough, so they can eventually limit the accuracy of the main task. To address this limitation, we design a joint formulation of these tasks, which yields mutual benefits and improves the accuracy on both tasks, compared with an individual task formulation.

Fig. 1.3 illustrates the three joint estimation cases, an overarching aim of this dissertation. Having two temporally consecutive frames as input and motion estimation as a common denominator, we define the joint objectives in the temporal domain.

(a) We first explore the joint estimation between semantic segmentation and optical flow. Each task provides an important cue to the other when two temporally consecutive frames are available. The semantic cue can help resolve matching ambiguity for optical flow estimation under the following conditions: *(i)* the matching correspondence should satisfy the same semantic class and *(ii)* static objects should follow the epipolar motion caused by camera ego-motion. Also, temporal correspondence from optical flow can encourage temporally consistent semantic segmentation. Joint estimation of optical flow and lower-level segmentation, via such as perceptual grouping from visual appearance and proximity, can also be an interesting formulation that strongly benefits both subjects. However, in this dissertation, we primarily focus on the connection of both tasks through high-level semantic knowledge.

(b) We then investigate a close relationship between optical flow and occlusion estimation. The interaction of both tasks is well established: temporal motion induces occlusion, and the occlusion is as an outlier for the matching problem. Thus, a more accurate estimate of one task can help the other and circle back. We demonstrate how the two tasks can mutually benefit each other in both traditional energy-based models and recent CNN-based models.

(c) Lastly, we look into the joint depth and 3D motion estimation, which is basically 3D scene flow estimation in a monocular setting. Estimation of 3D scene flow in a monocular setting is a highly ill-posed setup, where both scale and depth ambiguity

Figure 1.3: **Overview of our joint objectives**: In each chapter we present the following joint objectives, *(a)* joint optical flow and semantic segmentation, *(b)* joint optical flow and occlusion estimation, and *(c)* monocular scene flow.

exist. Exploiting CNNs, we demonstrate a monocular scene flow solution that can additionally output depth and optical flow as subtasks.

1.3 CONTRIBUTIONS AND OUTLINE

Here, we summarize the contributions of this dissertation along with a brief outline.

1.3.1 *Contributions*

Bridging optical flow and semantic segmentation. We present a joint optical flow and semantic segmentation approach in Chapter 3. We first introduce an accurate piecewise parametric optical flow formulation, which itself already outperforms previous work. Then we additionally apply the epipolar constraint for pixels that should be consistent with the camera ego-motion, as inferred by the semantic information. Also, we show that accurately estimated flow helps enforce temporal consistency on

the semantic segmentation. We successfully demonstrate that our joint formulation improves the accuracy of both tasks.

Joint optical flow and occlusion estimation. In Chapter 4, we address the mutual dependency between optical flow and occlusion map estimation and propose a joint energy formulation and optimization method. We exploit the two key symmetry properties of the optical flow field and occlusion map within the two consecutive images: forward-backward flow consistency and occlusion-disocclusion symmetry, which especially yields significant accuracy gains. We demonstrate how this joint, symmetric treatment combined with a piecewise rigid formulation allows optical flow estimation without post-processing.

We further explore this joint estimation of optical flow and occlusion using CNNs in Chapter 5. We propose an Iterative Residual Refinement (IRR) scheme that takes the output from a previous iteration as input and iteratively refines it, simply using a single network block with shared weights. We show that the weight-sharing design can significantly increase the accuracy without additional parameters, or even with fewer of them, depending on the backbone network. As in traditional energy-based formulation, we demonstrate that bi-directional optical flow and joint estimation with occlusion further improves the accuracy for both tasks.

Joint depth and 3D motion estimation. We propose joint depth and 3D scene flow estimation in a monocular camera setup in Chapter 6. We solve this ill-posed problem by taking an inverse problem view; we estimate scene flow in the monocular setting by decomposing a classical optical flow cost volume into scene flow and depth, using a CNN with a single joint decoder. We demonstrate that such approach indeed simplifies existing joint depth and flow estimation methods by yielding competitive accuracy, even with a simpler network. We train the networks in a self-supervised manner by introducing a self-supervised loss function as well as a suitable data augmentation, which resolves the issues related to the lack of 3D data annotation for training.

1.3.2 *Outline*

- In Chapter 2, we first start with introducing the background and related works. The literature review of optical flow in this chapter was published previously as Hur and Roth (2020a).
- Chapter 3 presents a joint optical flow and semantic segmentation approach, which was previously published as Hur and Roth (2016).
- Then in Chapter 4, we propose a joint energy formulation for optical flow and occlusion estimation, which exploits the symmetry between them. This technical part was published previously as Hur and Roth (2017).
- We further demonstrate a joint estimation of optical flow and occlusion using CNNs, in Chapter 5. This work was previously published as Hur and Roth (2019).

- Chapter 6 proposes a monocular scene flow estimation that jointly estimates depth and 3D scene flow in a monocular camera setup. This technical part was previously published as Hur and Roth (2020b).
- Lastly in Chapter 7, we conclude with a summary of our contributions and discussion on future challenges.

2

BACKGROUND AND RELATED WORK

CONTENTS

As briefly discussed in Chapter 1, we define our joint objectives in the temporal domain: jointly estimating motion with occlusion, depth, or semantic segmentation, where motion estimation serves as a basis for all joint objectives. Therefore, we first review the background on motion estimation: optical flow in Section 2.1, and scene flow in Section 2.2 respectively. Then, we review each of the individual objectives (*i. e.*, occlusion estimation, semantic segmentation, and depth estimation) along with the joint estimation with motion in Section 2.3.

2.1 OPTICAL FLOW

Influenced by the variational approach of Horn and Schunck (1981), earlier literature on optical flow estimation had been dominated by classical energy-based models, which formulate optical flow estimation as an energy minimization problem. Afterwards, as the practical benefits of CNNs over conventional methods have become apparent in numerous areas of computer vision and beyond, they have also seen increased adoption in the context of motion estimation to the point where the current state of the art in terms of accuracy is set by CNN approaches. In this section, we review both the classical energy-based models (in Section 2.1.1) and the current state of CNNs for optical flow estimation (in Section 2.1.2), in addition to how the transition happened in-between. Also we provide an overview of optical flow approaches based

on unsupervised or self-supervised learning (in Section 2.1.3), which tries to overcome the training data dependency issue that supervised learning based methods exhibit.

2.1.1 *Classical energy-based approaches*

Variational approach. For more than three decades, research on optical flow estimation has been heavily influenced by the variational approach of Horn and Schunck (1981). Their basic energy minimization formulation consists of a data term, which encourages brightness constancy between temporally corresponding pixels, and a spatial smoothness term, which regularizes neighboring pixels to have similar motion in order to overcome the aperture problem. The spatially continuous optical flow field $\mathbf{u} = (u_x, u_y)$ is obtained by minimizing

$$E(\mathbf{u}) = \int \left((I_x u_x + I_y u_y + I_t)^2 + \alpha^2 \left(\|\nabla u_x\|^2 + \|\nabla u_y\|^2 \right) \right) dx\, dy, \tag{2.1}$$

where I_x, I_y, I_t are the partial derivatives of the image intensity I with respect to x, y, and t. To minimize Eq. (2.1) in practice, spatial discretization is necessary. In such a spatially discrete form, the Horn and Schunck model (Horn and Schunck, 1981) can also be re-written in the framework of standard pairwise Markov Random Fields (MRFs) (Boykov et al., 1998; Li, 1994) through a combination of a unary data term $D(\cdot)$ and a pairwise smoothness term $S(\cdot, \cdot)$,

$$E(\mathbf{u}) = \sum_{\mathbf{p} \in \mathcal{I}} D(\mathbf{u_p}) + \sum_{\mathbf{p},\mathbf{q} \in \mathcal{N}} S(\mathbf{u_p}, \mathbf{u_q}), \tag{2.2}$$

where $\mathcal{I}$ is the set of image pixels and the set $\mathcal{N}$ denotes spatially neighboring pixels. Starting from this basic formulation, much research has focused on designing better energy models that more accurately describe the flow estimation problem (see Fortun et al. (2015) and Tu et al. (2019) for reviews of such methods).

Improving the data term. The brightness constancy assumption that the basic Horn and Schunck model (Horn and Schunck, 1981) relies on is often violated due to subtle brightness changes, shadow, occlusion or motion blur between two frames. This drawback had motivated further research on designing a better data term that can be robust under those circumstances.

Beyond the simple brightness constancy, imposing the gradient constancy (Brox et al., 2004) or high-order constancy constraints (Papenberg et al., 2006), such as Hessian or Laplacian, were proposed. Instead of per-pixel measure, there were several works that introduced patch-based similarity measures such as Normalized Cross Correlation (NCC) (Drulea and Nedevschi, 2013) or the Census transform (Stein, 2004) that captures the local context and thus is robust to illumination changes.

One problem of the quadratic penalty function by Horn and Schunck (1981) in Eq. (2.1) is that it is not robust to outliers. Thus, several previous works explored robust penalty functions such as the Charbonnier (Bruhn et al., 2005; Charbonnier et al., 1994), generalized Charbonnier (Sun et al., 2010a), or Lorentzian (Black and Anandan, 1996) that can reduce the large penalty from the outliers.

Improving the regularization. In order to compensate the data term design that is not robust to illumination changes or occlusion, various types of regularization strategies have been studied. As in Eqs. (2.1) and (2.2), the pairwise smoothness term is designed to encourage smooth motion while trying to preserve the motion discontinuity at motion boundaries (Weickert and Schnörr, 2001), combined with the robustness function (Black and Anandan, 1996; Mémin and Pérez, 1998) or total variation (Brox et al., 2004; Werlberger et al., 2010; Zach et al., 2007). Image-driven regularization approaches (Alvarez León et al., 1999; Alvarez et al., 2000; Xu et al., 2011) demonstrated encouraging the motion discontinuity based on the image gradient, which is inspired by the fact that the object boundary often corresponds to the motion boundary. Beyond simply applying smoothness between neighboring pixels, non-local regularization approaches (Drulea and Nedevschi, 2013; Krähenbühl and Koltun, 2012; Ranftl et al., 2014; Sun et al., 2010a) considered longer range connection between pixels by calculating the affinity between pixels within a certain range.

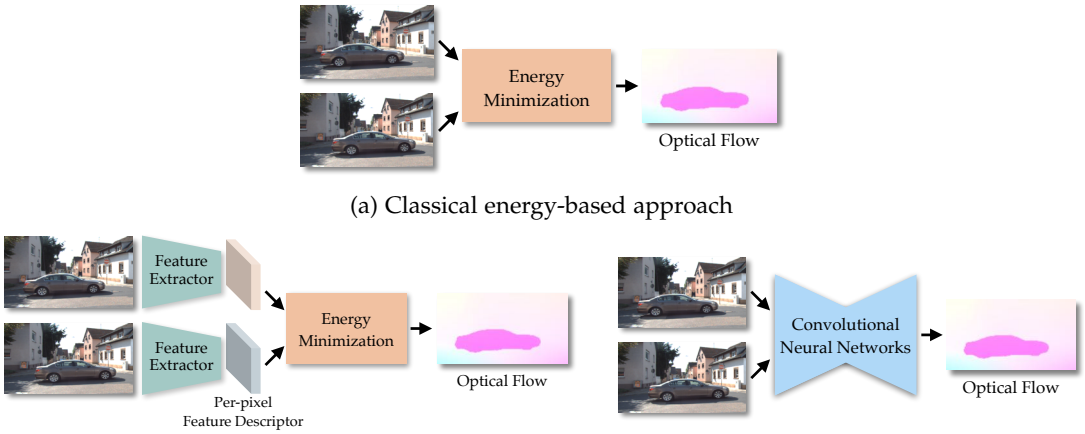
Piecewise parametric model. Piecewise parametric approaches using a homography model have demonstrated a promising direction for regularizing the motion field by taking geometric cues into account. Representing the scene as a set of planar surfaces significantly reduces the number of unknowns; at the same time, parameterizing the motion of surfaces by 8-DoF or 9-DoF transforms ensures sufficient diversity and generality of their motion (Hornáček et al., 2014; Menze and Geiger, 2015; Vogel et al., 2014; 2013b; 2015; Yang and Li, 2015). Unlike in the stereo setting (Vogel et al., 2014; 2013b; 2015) where 3D depth cue can be obtained, the monocular setup (*i. e.*, two temporally consecutive images) makes the problem more challenging; hence the type of regularization becomes much more crucial. Hornáček et al. (2014) introduced a 9-DoF plane-induced model for optical flow via continuous optimization. Their method shows its strength on rigid motions, but is weaker on poorly textured regions because of the lack of global support. Yang and Li (2015) instead used a 8-DoF homography motion in 2D space with adaptive size and shape of the pieces via discrete optimization.

In Chapter 4, we also introduce a piecewise parametric optical flow model that relies on an 8-DoF parameterization with occlusion reasoning, which shows competitive accuracy, comparing to previous work.

2.1.2 *CNN-based approaches*

2.1.2.1 *CNNs as feature extractor*

The relatively recent success of applying CNNs with backpropagation on large-scale image classification tasks (Krizhevsky et al., 2012) paved the way for applying CNNs to various other computer vision problems, including optical flow as well. Early work that applied CNNs to optical flow used them as an advanced feature extractor (Bai et al., 2016; Bailer et al., 2017; Gadot and Wolf, 2016; Güney and Geiger, 2016), as sketched in Fig. 2.1b. The main idea behind this is to substitute the data term (*e. g.*, in



(a) Classical energy-based approach

(b) Using CNNs as a feature extractor

(c) CNN regression architecture

Figure 2.1: Transition from *(a)* classical energy-based approaches to *(b)* CNN-based approaches that use CNNs as a feature extractor or to *(c)* end-to-end trainable CNN regression architectures.

Eqs. (2.1) and (2.2)) in classical energy-based formulations with a CNN-based feature matching term. Instead of using image intensities, image gradients, or other hand-crafted features as before, CNNs enable learning feature extractors such that each pixel can be represented with a high-dimensional feature vector that combines a suitable amount of distinctiveness and invariance, for example to appearance changes. The putative similarity between regions is given by the feature distance. The remaining pipeline, including using the smoothness term as well as the optimization strategies, remain the same.

Gadot and Wolf (2016) proposed a method called PatchBatch, which was among the first flow approaches to adopt CNNs for feature extraction. PatchBatch (Gadot and Wolf, 2016) is based on a Siamese CNN feature extractor that is fed 51×51 input patches and outputs a 512-dimensional feature vector using a shallow 5-layer CNN. Then, PatchBatch adopts Generalized PatchMatch (Barnes et al., 2010) as an Approximate Nearest Neighbor (ANN) algorithm for correspondence search, *i. e.*, matching the extracted features between two images. The method constructs its training set by collecting positive corresponding patch examples given ground-truth flow and negative non-matching examples by randomly shifting the image patch in the vicinity of where the ground-truth flow directs. The intuition of collecting negative examples in such a way is to train CNNs to be able to separate non-trivial cases and extract more discriminative features. The shallow CNNs are trained using a variant of the DrLIM (Hadsell et al., 2006) loss, which minimizes the squared L_2 distance between positive patch pairs and maximizes the squared L_2 distance between negative pairs above a certain margin.

In a similar line of work, Bailer et al. (2017) proposed to use the thresholded hinge embedding loss for training the feature extractor network. The hinge embedding loss based on the L_2 loss function has been commonly used to minimize the feature distance between two matching patches and to maximize the feature distance above a

certain threshold between non-matching patches. However, minimizing the L_2 loss of some challenging positive examples (*e. g.*, with appearance difference or illumination changes) can move the decision boundary into an undesired direction and lead to misclassification near the decision boundary. Thus, Bailer et al. (2017) proposed to use another threshold on the hinge embedding loss in order to prevent the network from minimizing the L_2 distance too aggressively, which has led to more accurate flow estimates.

Meanwhile, Güney and Geiger (2016) demonstrated successfully combining a CNN feature matching module with a discrete Maximum A Posteriori (MAP) estimation approach based on a pairwise MRF model. The proposed CNN module outputs per-pixel descriptors, from which a cost volume is constructed by calculating feature distances between sample matches. This is input to a discrete MAP estimation approach (Menze et al., 2015a) to infer the optical flow. Bai et al. (2016) followed a similar setup overall, but utilized Semi-Global block Matching (SGM) (Hirschmüller, 2008) to regress the output optical flow from the cost volume, which is constructed by calculating a distance between features from CNNs.

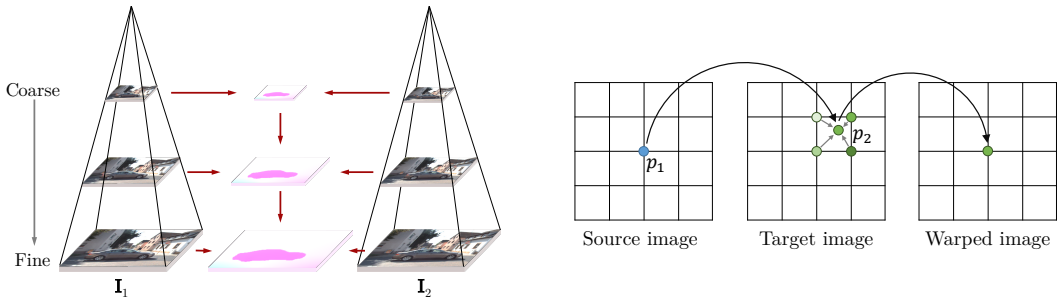
Taken together, these approaches have successfully demonstrated that the benefits of the representational power of CNNs can be combined with well-proven classical energy-based models. Specifically, they demonstrated more accurate estimates on inliers and more precise estimates on object boundaries than previous baselines with hand-constructed features.

2.1.2.2 *End-to-end regression architectures*

Concurrently with the development of feature extraction-based networks, active research also started on developing end-to-end CNN architectures for optical flow estimation based on regression, as sketched in Fig. 2.1c. Unlike methods that use CNNs only for feature extraction as addressed above, such regression methods exploit CNNs for the entire pipeline and directly output optical flow from a pair of input images. By substituting classical regularizers and avoiding energy minimization, these CNN-based methods combine the advantages of end-to-end trainability and runtime efficiency.

Dosovitskiy et al. (2015) proposed the first end-to-end CNN architecture for estimating optical flow, called FlowNet, which has two main architectural lines, **FlowNetS** and **FlowNetC**. The two models are fundamentally based on an hourglass-shaped neural network architecture that consists of an encoder and a decoder, and differs only in the encoder part. The input of FlowNetS is just a concatenation of a pair of input images, while FlowNetC first extracts a feature map for each input image using a shared encoder and then constructs a cost volume with correlation operations, which is then fed into the subsequent network layers.

To train the networks in a supervised way, Dosovitskiy et al. (2015) established a synthetic dataset (called FlyingChairs) in order to overcome the shortage of suitable training data. Due to the intrinsic differences between synthetic and real-world images, however, unfortunately FlowNet trained on the synthetic domain does not generalize well to real images, being outperformed by classical energy-based methods.



(a) Coarse-to-fine estimation

(b) Backward warping

Figure 2.2: *(a)* The classical coarse-to-fine concept proceeds by estimating optical flow using a multi-scale image pyramid, starting from the coarsest level to the finest level. By gradually estimating and refining optical flow through the pyramid levels, this approach can handle large displacements better and improve accuracy. *(b)* Backward warping is commonly used in optical flow estimation. For each pixel p_1 in the source image, the warped image obtains the intensity from (sub)pixel location p_2, which is obtained from the estimated flow. Bilinear interpolation is often used to obtain the pixel intensity at the non-integer coordinate.

Yet, importantly, FlowNet demonstrated the possibility of employing an end-to-end regression architecture for optical flow estimation. Moreover, FlowNet established several standard practices for training optical flow networks such as learning rate schedules, basic network architectures, data augmentation schemes, and the necessity of pre-training on synthetic datasets, which have substantially impacted follow-up research.

Ranjan and Black (2017) proposed **SPyNet**, which incorporates the classical "coarse-to-fine" concept (please refer to Fig. 2.2a for an illustration) into a CNN model and updates the residual flow over multiple pyramid levels. SPyNet consists of 5 pyramid levels, and each pyramid level consists of a shallow CNN that estimates flow between a source image and a target image, which is warped by the current flow estimate (see Fig. 2.2b). This estimate is updated so that the network can residually refine optical flow through a spatial pyramid and possibly handle large displacements. Compared to FlowNet, SPyNet significantly reduces the number of model parameters by 96% by using a pyramid-shaped architecture, while achieving comparable and sometimes even better results than FlowNet.

Meanwhile, Ilg et al. (2017) proposed **FlowNet2**, which significantly improves the flow accuracy over their previous FlowNet architecture and started to outperform classical energy-based approaches. To overcome the limitations of FlowNet (*e. g.*, blurry outputs and lower accuracy in general), Ilg *et al.* proposed the key idea that by stacking multiple FlowNet-style networks, one can sequentially refine the output from the previous network modules. Despite of the conceptual simplicity, stacking multiple networks is very powerful and significantly improves the flow accuracy by more than 50% over FlowNet. Additionally, Ilg *et al.* revealed several important practices for training their networks, including the necessity of pre-training on synthetic datasets

and fine-tuning on real image datasets, the effectiveness of using a correlation layer, and the guidance of proper learning rate schedules.

After the successful demonstration of FlowNet2 (Ilg et al., 2017) that end-to-end regression architectures can outperform energy-based approaches, further investigations on finding better network architectures have continued. Sun et al. (2018) proposed an advanced architecture called **PWC-Net** by exploiting well-known design principles from classical approaches. PWC-Net relies on three main design principles: *(i)* pyramid, *(ii)* warping, and *(iii)* cost volume. Similar to SPyNet (Ranjan and Black, 2017), PWC-Net estimates optical flow in a coarse-to-fine way with several pyramid levels, but PWC-Net constructs a feature pyramid by using CNNs, instead of an image pyramid as in SPyNet. Next, PWC-Net constructs a cost volume with a feature map from the source image and the warped feature map from the target image based on the current flow. Then, the subsequent CNN modules act as a decoder that outputs optical flow from the cost volume. In terms of both accuracy and practicality, PWC-Net (Sun et al., 2018) set a new state of the art with its light-weight architecture allowing for shorter training times, faster inference, and more importantly, clearly improved accuracy. Comparing to FlowNet2 (Ilg et al., 2017), PWC-Net is 17 times smaller in model size and twice as fast during inference while being more accurate. Similar to SPyNet, the computational efficiency stems from using coarse-to-fine estimation, but PWC-Net crucially demonstrates that constructing and warping feature maps instead of using downsampled warped images yields much better accuracy.

Concurrently, **LiteFlowNet** (Hui et al., 2018) also demonstrated utilizing a multi-level pyramid architecture that estimates flow in a coarse-to-fine manner, proposing another light-weight regression architecture for optical flow. The major technical differences to PWC-Net are that LiteFlowNet residually updates optical flow estimates over the pyramid levels and proposes a flow regularization module. The proposed flow regularization module creates per-pixel local filters using CNNs and applies the filters to each pixel so that customized filters refine flow fields by considering neighboring estimates. The regularization module is given the optical flow, feature maps, and occlusion probability maps as inputs to take motion boundary information and occluded areas into account in creating per-pixel local filters. The experimental results demonstrate clear benefits, especially from using the regularization module that smoothes the flow fields while effectively sharpening motion boundaries, which reduces the error by more than 13% on the training domain.

Yin et al. (2019) proposed a general probabilistic framework termed **HD**^{3} for dense pixel correspondence estimation, exploiting the concept of the so-called match density, which enables the joint estimation of optical flow and its uncertainty. Mainly following the architectural design of PWC-Net, the method estimates the full match density in a hierarchical and computationally efficient manner. The estimated spatially discretized match density can then be converted into optical flow vectors while providing an uncertainty assessment at the same time, which is rather different from all previous regression networks above. On established benchmarks datasets, their experimental results demonstrate state-of-the-art accuracy regarding both optical flow and uncertainty measures.

Table 2.1: Overview of the main technical design principles of end-to-end optical flow architectures: FlowNetS (Dosovitskiy et al., 2015), SPyNet (Ranjan and Black, 2017), FlowNet2 (Ilg et al., 2017), PWC-Net (Sun et al., 2018), LiteFlowNet (Hui et al., 2018), HD^3 (Yin et al., 2019), VCN (Yang and Ramanan, 2019), and RAFT (Teed and Deng, 2020).

Methods	FlowNetS	SPyNet	FlowNet2	PWC-Net	LiteFlowNet	HD^3	VCN	RAFT
Pyramid	–	5-level image	3-level feature	6-level feature	6-level feature	5-level feature	6-level feature	4-level feature
Warping	–	Image	Image	Feature	Feature	Feature	Feature	–
Cost volume	–	–	3D	3D	3D	3D	4D	4D
Network stacking	–	–	5	–	–	–	–	–
Flow inference	Direct	Residual	Direct	Direct	Residual	Residual	Hypothesis selection	Residual with GRU
Parameters (M)	38.67	1.20	162.49	8.75	5.37	39.6	6.20	5.3

While the cost volume has been commonly used in backbone architectures (Dosovitskiy et al., 2015; Hui et al., 2018; Ilg et al., 2017; Sun et al., 2018; Yin et al., 2019), its representation is mainly based on a heuristic design. Instead of representing the matching costs between all pixels (x, y) with their possible 2D displacements (u, v) into a 4D tensor (x, y, u, v), the conventional design is based on a 3D cost volume – a 2D array (x, y) augmented with a uv channel, which is computationally efficient but often yields limited accuracy and overfitting. To overcome this limitation, Yang and Ramanan (2019) proposed Volumetric Correspondence Networks (**VCN**), which are based on true 4D volumetric processing: constructing a proper 4D cost volume processing with an approximated 4D convolution kernels for computational efficiency. Through proper 4D volumetric processing, the method further pushes both accuracy and practicality on widely used public benchmarks, improving generalization and demonstrating faster training convergence.

Afterward, Teed and Deng (2020) introduced a newer generation of deep network architecture for optical flow, called Recurrent All-Pairs Field Transforms (**RAFT**). RAFT first pre-computes a multi-scale 4D correlation volume for all possible pairs of pixels and then iteratively updates optical flow only at a single high resolution using a Gated Recurrent Unit (GRU) that performs lookups on the pre-computed cost volume. Unlike previous work (Hui et al., 2018; Sun et al., 2018; Yang and Ramanan, 2019) based on the coarse-to-fine estimation, the operation only at the high resolution can bring a benefit of keeping the fine details for small objects that are sometimes lost at the coarser level. Possible demerits from the single scale estimation such as limited search space or heavy computational cost are overcome by computing the cost volume for all possible pairs but only once. RAFT not only sets a new state of the art on established public benchmark datasets such as MPI Sintel and KITTI, but also demonstrated strong generalization to unseen domains.

Table 2.2: Quantitative comparison on public benchmarks: MPI Sintel (Butler et al., 2012) and KITTI (Geiger et al., 2012; Menze and Geiger, 2015).

Methods	MPI Sintel [a]		KITTI [b]	
	Clean	Final	2012	2015
FlowNetS (Dosovitskiy et al., 2015)	6.158	7.218	37.05%	–
SPyNet (Ranjan and Black, 2017)	6.640	8.360	12.31%	35.07%
FlowNet2 (Ilg et al., 2017)	3.959	6.016	4.82%	10.41%
PWC-Net (Sun et al., 2018)	4.386	5.042	4.22%	9.60%
LiteFlowNet (Hui et al., 2018)	3.449	5.381	3.27%	9.38%
HD^3 (Yin et al., 2019)	4.788	4.666	**2.26%**	6.55%
VCN (Yang and Ramanan, 2019)	2.808	4.404	–	6.30%
RAFT (Teed and Deng, 2020)	**1.609**	**2.855**	–	**5.10%**

[a] Evaluation metric: end point error (EPE).

[b] Evaluation metric: outlier rate (*i. e.*, less than 3 pixel or 5% error is considered an inlier)

Table 2.1 summarizes the main differences in technical design of the various end-to-end optical flow architectures discussed above. Starting from FlowNetS (Dosovitskiy et al., 2015), the methods are listed in chronological order. Table 2.2 compares the quantitative results of each method on the MPI Sintel (Butler et al., 2012) and KITTI benchmarks (Geiger et al., 2012; Menze and Geiger, 2015). Each method is pre-trained on synthetic datasets first and then fine-tuned on each benchmark. Looking at the two tables, we can gain some insights into which design choices lead to the observed accuracy improvements. First, having a pyramid structure by adopting a "coarse-to-fine" strategy makes networks more compact and improves the flow estimation accuracy (*e. g.*, from FlowNet (Dosovitskiy et al., 2015) to SPyNet (Ranjan and Black, 2017), PWC-Net (Sun et al., 2018), and LiteFlowNet (Hui et al., 2018)). Second, stacking networks can also improve the flow accuracy while linearly increasing the number of parameters (*e. g.*, from FlowNet (Dosovitskiy et al., 2015) to FlowNet2 (Ilg et al., 2017)). Third, constructing a cost volume by calculating a patch-wise correlation between two feature maps has become a standard approach and is more beneficial than not using it (*e. g.*, SPyNet vs. PWC-Net). Fourth, even if based on similar conceptual designs, subtle design differences or additional modules can further lead to accuracy improvements (*e. g.*, LiteFlowNet (Hui et al., 2018) vs. PWC-Net (Sun et al., 2018)). Lastly, research on better fundamental designs such as the way of processing output (*e. g.*, the recurrent update using GRU and a cost-volume lookup table (Teed and Deng, 2020)) or the cost volume representation (*e. g.*, 4D cost volume (Teed and Deng, 2020; Yang and Ramanan, 2019)) can lead to further improvement, sometimes quite significantly so.

Fig. 2.3 shows a qualitative comparison of each method on an example from the Sintel Final Test set (Butler et al., 2012). The optical flow visualizations and the error maps demonstrate how significantly end-to-end methods have been improved over the past few years, especially near motion boundaries and in non-textured areas.

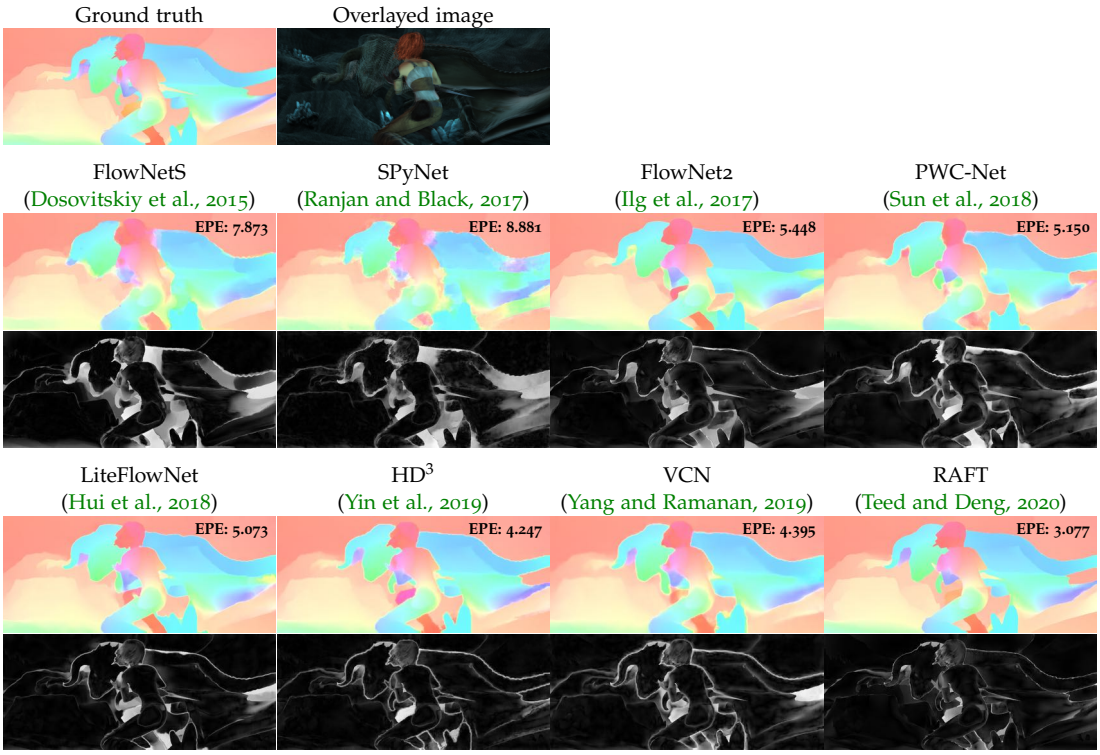


Figure 2.3: **Qualitative comparison of end-to-end architectures**: Example from Sintel Final Test (Butler et al., 2012). The first column shows the ground-truth flow and the overlayed input images. In the further columns, we show the color-coded flow visualization of each method, overlayed with the EPE and their error maps (the brighter a pixel, the higher its error).

2.1.2.3 *Add-on modules for accuracy improvement*

Along with the advances of the end-to-end regression architectures, there has been a parallel line of works that propose generic add-on modules that can be possibly applied to any backbone network architectures and improve the accuracy.

Feature warping has demonstrated as one of the essential technical components in a series of works (Hui et al., 2018; Sun et al., 2018; Yang and Ramanan, 2019), yet so-called "ghosting effect" has been overlooked. When backward-warping feature maps at the target frame to the source frame (*cf*. Fig. 2.2b), feature maps at disoccluded areas at the target frame are duplicated and thus create matching ambiguity. Zhao et al. (2020) proposed an asymmetric occlusion-aware feature matching module that learns to estimate a rough occlusion mask without explicit supervision and filters out the warped feature map in the occluded areas so that the doubled feature map does not affect the matching process. By adopting into PWC-Net (Sun et al., 2018), the method improves the optical flow accuracy by 19% in MPI Sintel and 34% in the KITTI benchmark over PWC-Net with reasonable occlusion maps as a by-product.

Xiao et al. (2020) demonstrated a learnable cost volume using an elliptical inner product, which generalizes the standard Euclidean inner product by a symmetric and positive definite kernel matrix. From the generalization using the Cayley rep-

resentation, the learnable cost volume is able to have more representation capacity because it calculates correlation among different channel dimensions and weighs each dimension differently. By plugging into several backbone architectures (Liu et al., 2019b; Sun et al., 2018; Yang and Ramanan, 2019) and replacing the vanilla cost volume, the experiment demonstrates consistent accuracy improvements as well as robustness against the adversarial perturbations over the baselines.

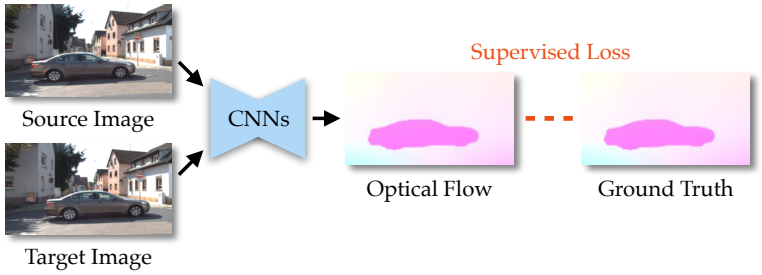
Typical encoder-decoder networks tend to produce blurred estimation as well as boundary artifacts due to the strong reduction of spatial resolution. To address this issue, Wannenwetsch and Roth (2020) introduced probabilistic pixel-adaptive convolutions that utilize both image guidance data and the confidence estimates to refine outputs of off-the-shelf methods. As a post-processing module for optical flow and semantic segmentation as well, Wannenwetsch and Roth (2020) successfully demonstrated a clear reduction in boundary artifacts and improved the accuracy as well.

Hui and Loy (2020) demonstrated two add-on modules, cost volume modulation and flow field deformation, which further improves the accuracy of off-the-shelf methods, FlowNetC (Dosovitskiy et al., 2015), PWC-Net (Sun et al., 2018), and LiteFlowNet (Hui et al., 2018). The cost volume modulation module applies a learned affine transform to the cost volume to filter outliers that may exist due to matching ambiguity or occlusion. The flow field deformation module learns to adopt the neighboring estimated flow so that an inaccurate estimate is replaced by an accurate estimate nearby.

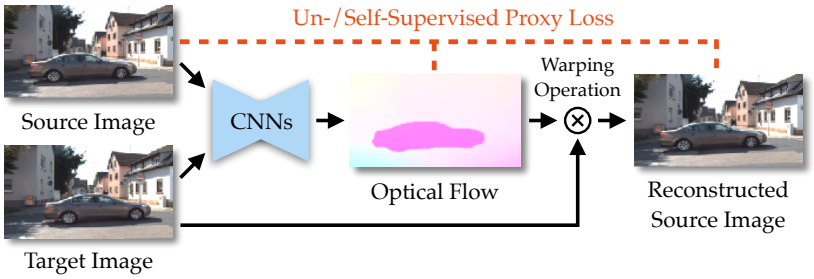
Hofinger et al. (2020) also proposed several technical components that improves the optical flow CNN architectures based on the pyramid structure and cost volume. Analogous to Teed and Deng (2020), Hofinger et al. (2020) proposed to use a sampling-based cost volume construction instead of warping-based way in order to avoid the ghosting effect and preserve fine details. Furthermore, they demonstrated a gradient stopping strategy between pyramid levels as well as distillation concept for sequentially training the model on multiple datasets, which overall improves the final result. The proposed technical components are successfully validated on PWC-Net (Sun et al., 2018) and HD^3 (Yin et al., 2019).

Subtle design choices in CNN architecture matter, but so do training details. Bar-Haim and Wolf (2020) took a close look into the augmentation scheme and demonstrate that certain cropping and scaling augmentation protocol yields imbalanced sampling bias and eventually degrades the accuracy. By improving the sampling strategy (*e. g.*, using a larger crop size and a careful crop positioning so that it does not neglect challenging samples), the method improved the accuracy by more than 10% on MPI Sintel and 12% on KITTI benchmark datasets.

In Chapter 5, we propose an Iterative Residual Refinement (IRR) scheme based on weight sharing that can be combined with several backbone networks (Dosovitskiy et al., 2015; Sun et al., 2018). Inspired by classical energy minimization-based methods, our model estimates both optical flow and occlusion by iteratively using a weight-shared network, which reduces the number of parameters, improves the accuracy, or even achieves both. Further integrating bi-directional flow estimation into the IRR scheme further boosts the accuracy, outperforming state-of-the-art results for both



(a) Supervised loss



(b) Unsupervised or self-supervised proxy loss

Figure 2.4: **Comparison of loss function in supervised learning and unsupervised (or self-supervised) learning of optical flow**: *(a)* For supervised learning, the loss is directly applied to output with given ground truth. *(b)* For un-/self-supervised learning, a proxy loss function is designed and indirectly applied to output.

optical flow and occlusion estimation across several standard datasets (Butler et al., 2012; Menze and Geiger, 2015).

2.1.3 *Unsupervised/self-supervised learning*

Aside from the question of how to design deep network architectures for optical flow estimation, another problem dimension has grown into prominence recently – how to train such CNNs for optical flow especially in the context of the limited quantities of ground-truth data available in practice. Most (early) CNN approaches are based on standard supervised learning and utilize synthetically generated data. While synthetic datasets enable training CNNs with a large amount of labeled data, the networks only trained on synthetic datasets perform relatively poorly on real-world datasets due to the domain mismatch between the training domain and the target domain. As just discussed, supervised approaches thus require fine-tuning on the target domain for better accuracy.

However, this can be problematic if there is no ground truth optical flow available for the target domain. To resolve this issue, unsupervised learning approaches have been proposed to directly train CNNs on the target domain without having access to any ground truth flow. Such methods are also called self-supervised, as the supervisory signal comes from the input images themselves. Fig. 2.4 compares how the loss function is applied between supervised and un-/self-supervised learning

case. Unlike the supervised learning where the loss is directly applied to the output (Fig. 2.4a), a proxy loss function is designed and indirectly applied to the flow output in case of un-/self-supervised learning (Fig. 2.4b). In this section, we will overview existing unsupervised or self-supervised learning methods and discuss how they have progressed to achieve results that are competitive with many supervised methods.

Ahmadi and Patras (2016) pioneered unsupervised learning-based optical flow using CNNs. Inspired by the classical Horn and Schunck (Horn and Schunck, 1981) method, Ahmadi and Patras (2016) used the classical optical flow constraint equation as a loss function for training the network. By minimizing this unsupervised loss function, the network learns to predict optical flow fields that satisfy the optical flow constraint equation on the input images, *i. e.*, the brightness constancy assumption. By demonstrating that the flow accuracy is close to the best supervised method at the time, *i. e.* FlowNet (Dosovitskiy et al., 2015), Ahmadi and Patras (2016) suggest that unsupervised learning of networks for optical flow estimation is possible and can overcome some of the limitations of supervised learning approaches.

Concurrently, Yu et al. (2016) and Ren et al. (2017a) proposed to use a proxy unsupervised loss that is inspired by a standard MRF formulation. Following classical concepts, the proposed unsupervised proxy loss consists of a data term and a smoothness term as in Eq. (2.2). The data term directly minimizes the intensity difference between the first image and the warped second image from estimated optical flow, and the smoothness term penalizes flow differences between neighboring pixels. Both methods demonstrate that directly training on a target domain (*e. g.*, the KITTI datasets (Geiger et al., 2012)) in an unsupervised manner performs competitive to or sometimes even outperforms the same network that is trained on a different domain (*e. g.*, the FlyingChairs dataset (Dosovitskiy et al., 2015)) in a supervised manner. This observation suggests that unsupervised learning approaches can be a viable alternative to supervised learning if labeled data for training is not available in the target domain.

In follow-up work, Zhu and Newsam (2017) showed that the backbone network can be improved by using dense connectivity. By adopting dense blocks (Huang et al., 2017) with skip connections between all convolutional layers, Zhu and Newsam (2017) improve the flow accuracy by more than 10% on public benchmark datasets over Yu et al. (2016) on average, which uses FlowNet (Dosovitskiy et al., 2015) as a backbone network. This indicated the importance of choosing the right backbone network in the unsupervised learning setting as well.

Zhu et al. (2017c) also proposed a different direction of unsupervised learning, combining an unsupervised proxy loss and a guided supervision loss using proxy ground truth obtained from an off-the-shelf classical energy-based method. In the circumstance that learning with the unsupervised proxy loss is outperformed by the classical energy-based method, the guided loss can help and even achieve better accuracy than either of the two losses alone.

Unsupervised or self-supervised learning of optical flow relies on minimizing a proxy loss rather than estimating optical flow close to some ground truth. Thus, designing a faithful proxy loss is critical to its success. Meister et al. (2018) proposed a proxy loss function that additionally considers occlusions, demonstrates better

accuracy than previous unsupervised methods, and outperforms the supervised backbone network (*i.e.*, FlowNet (Dosovitskiy et al., 2015)). Further, bi-directional flow is estimated from the same network by only switching the order of input images and occlusions are detected using a bi-directional consistency check. The proxy loss is applied only to non-occluded regions as the brightness constancy assumption does not hold for occluded pixels. In addition, Meister et al. (2018) suggested using a higher-order smoothness term and a ternary census loss (Stein, 2004; Zabih and Woodfill, 1994) to obtain a data term that is robust to brightness changes. This advanced proxy loss significantly improves the accuracy by halving the error compared to previous unsupervised learning approaches. The approach of Meister et al. (2018) results in better accuracy than supervised approaches pre-trained on synthetic data alone (assuming the same backbone), which suggests that directly training on the target domain in an unsupervised manner can be a good alternative to supervised pre-training with synthetic data.

Wang et al. (2018) also introduced an advanced proxy loss that takes occlusion into account and is applied only to non-occluded regions. Similar to Meister et al. (2018), Wang et al. (2018) estimate bi-directional optical flow and then obtain an occlusion mask for the forward motion by directly calculating disocclusion from the backward flow. They exploit the fact that occlusion from the forward motion is the inverse of disocclusion from the backward motion. Wang et al. (2018) improved the accuracy overall by 25% on public benchmark datasets compared to the unsupervised approach of Yu et al. (2016) and demonstrated good occlusion estimation results, close to those of classical energy-based approaches.

Janai et al. (2018) extended unsupervised learning of optical flow to a multi-frame setting, taking in three consecutive frames and jointly estimating an occlusion map. Based on the PWC-Net (Sun et al., 2018) architecture, they estimate bi-directional flow from the reference frame and occlusion maps for both directions as well. A basic unsupervised loss consisting of photometric and smoothness terms is applied only on non-occluded regions for estimating flow, and a constant velocity constraint is also used, which encourages the magnitude of forward flow and backward flow to be similar but going in opposite directions. Their experimental results demonstrate the benefits of using multiple frames, outperforming all two-frame-based methods with competitive accuracy of occlusion estimation against classical energy-based methods.

Liu et al. (2019b,c) demonstrated another direction for unsupervised (or self-supervised) learning by using a data distillation framework with student-teacher networks. Their two methods, DDFlow (Liu et al., 2019b) and its extension SelFlow (Liu et al., 2019c), distill reliable predictions from a teacher network, which is trained in an unsupervised manner (Meister et al., 2018), and use them as pseudo ground truth for training the student network, which is used at inference time. DDFlow (Liu et al., 2019b) proposed to randomly crop the predicted flow map from the teacher network as well as the input images, and then use them as pseudo ground truth to train the student network. The main intuition is that its reliably predicted optical flow from the non-occluded pixels in the teacher network can work as reliable pseudo ground truth for occluded pixels in the student network. SelFlow (Liu et al., 2019c) suggested a better data distillation strategy by exploiting superpixel knowledge

and hallucinating occlusions in non-occluded regions. Given the prediction from the teacher network, SelFlow (Liu et al., 2019c) superpixelizes the target frame and perturbs random superpixels by injecting random noise as if non-occluded pixels in the target images were occluded by randomly looking superpixels. Then likewise, those non-occluded pixels with reliable predictions from the teacher network become occluded pixels when training the student network, guiding to estimate reliable optical flow in occluded areas. Evaluating on public benchmark datasets, both SelFlow (Liu et al., 2019c) and DDFlow (Liu et al., 2019b) improved the accuracy over the previous works, suggesting a promising direction for self-supervised learning.

Based on the previous work using the distillation pipeline (Liu et al., 2019c), Liu et al. (2020) demonstrate a more effective data distillation scheme using data augmentation on the pseudo ground truth. On the input images and pseudo ground truth obtained from the teacher networks, Liu et al. (2020) apply photometric and geometric augmentation (*e. g.*, random crop, flip, zoom, and affine transform) in order to provide more diverse supervision to the student model. Liu et al. (2020) demonstrated that this regularization concept brought up to 13% accuracy boost in average on public benchmark datasets.

Unsupervised or self-supervised methods have demonstrated promising results over supervised methods. The success of these methods often relies on careful designs of the proxy loss function as well as training strategies. In this regard, Jonschkowski et al. (2020) systematically analyzed a set of key components of the design choices (*e. g.*, the data term, smoothness term, occlusion handling, input resolution, and augmentation etc.) and demonstrated which combination yields the best accuracy. Besides, Jonschkowski et al. (2020) further introduced several novel technical contributions such as cost volume normalization, applying smoothness before upsampling the flow map, and continual self-supervision with distillation. The combination of all these components in the end outperforms previous unsupervised approaches and performs on par with a supervised method, FlowNet2 (Ilg et al., 2017).

2.1.4 *Training datasets and their importance*

Supervised learning. As discussed in Section 2.1.2.2, almost all the supervised optical flow methods follow the same conventional training and evaluation protocols that are settled by Dosovitskiy et al. (2015) and Ilg et al. (2017). The methods first pre-trains their models on synthetic datasets (*e. g.*, FlyingChairs (Dosovitskiy et al., 2015) and FlyingThings3D (Mayer et al., 2016)), and then they fine-tune the models and evaluate on a target domain dataset (*e. g.* Middlebury (Baker et al., 2011), MPI Sintel or KITTI). The main reason of pre-training on synthetic datasets is because not only there exists no large-scale real-world dataset, but also models pre-trained on synthetic datasets tend to generalize well to the real-world domains. Of course, details on a synthetic data generation process matter for a generalization to the real-world domain; realism of rendered synthetic images (Sun et al., 2021) as well as matched motion statistics (Mayer et al., 2016) to a target dataset are the two well-known major factors. Sun et al. (2021) provides an analysis on which rendering details are crucial and proposes an

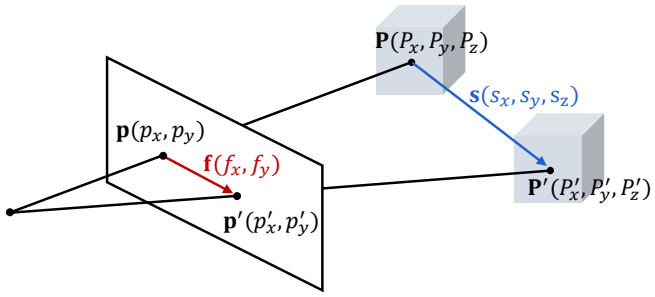


Figure 2.5: The relationship between scene flow and optical flow.

automatic synthetic data generation approach that optimizes the accuracy of a model on a target dataset.

For fine-tuning, early approaches (Dosovitskiy et al., 2015; Ilg et al., 2017; Sun et al., 2018) used annotated data only from a target domain dataset. However, later approaches (Sun et al., 2021; Teed and Deng, 2020; Zhang et al., 2021) demonstrate that using a mixture of annotated data results in better accuracy especially when there is not enough annotated data on the target domain (*e. g.* KITTI).

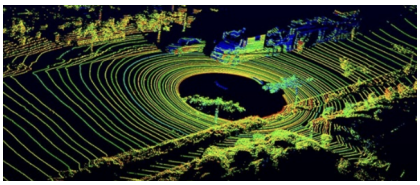
Self-supervised learning. Many self-supervised optical flow approaches have been proposed, but they only demonstrate that self-supervised learning can be a good substitute for supervised pre-training on synthetic datasets. Still, their accuracy is outperformed by the same model fine-tuned even with few-hundreds annotated examples after the pre-training. This limitation hinders the benefit of exploiting a large amount of unlabeled datasets and potentially eliminating the need of annotated data in target domains. The main reason is due to proxy loss function designs that determine the success of self-supervised methods. Once a proxy loss (or task) design that can outperform the accuracy of supervised fine-tuned models is found, self-supervised methods can fully benefit from the self-supervised learning and even can build a universal model that generalizes well to multiple target domains through multi-dataset training.

2.2 SCENE FLOW

Scene flow is commonly defined as a dense 3D motion field for each point in the scene and was first introduced by Vedula et al. (1999, 2005). Fig. 2.5 illustrates the relationship between optical flow and scene flow. When an object moves from a point $\mathbf{P}(P_x, P_y, P_z)$ to another point $\mathbf{P}'(P'_x, P'_y, P'_z)$ in 3D space, scene flow $\mathbf{s}(s_x, s_y, s_z)$ is the 3D motion between the two points (*i. e.*, $\mathbf{s} = \mathbf{P}' - \mathbf{P}$). Observed in the image coordinate, optical flow $\mathbf{f} = (f_x, f_y)$ is the apparent motion between the two pixels $\mathbf{p} = (p_x, p_y)$ in the reference image and $\mathbf{p}' = (p'_x, p'_y)$ in the target image (*i. e.*, $\mathbf{f} = \mathbf{p}' - \mathbf{p}$). In other words, optical flow is the projection of scene flow into the image coordinate.

The problem formulation of scene flow estimation varies depending on the type of input data. In case of stereo (*cf*. Fig. 2.6a) or monocular image sequences, each

(a) A pair of rectified stereo images (Geiger et al., 2012)



(b) 3D point cloud

(c) RGB-D image (Quiroga et al., 2014)

Figure 2.6: **A different type of input data for scene flow estimation**: *(a)* A pair of rectified images captured from a stereo camera, *(b)* RGB-D (Red, Green, Blue, and Depth) image, and *(c)* 3D point cloud captured from LiDAR (Light Detection and Ranging).

3D point can be estimated by disparity matching or scene prior knowledge learning, which often requires complex formulations. In case of 3D point cloud (*cf*. Fig. 2.6b) or RGB-D data (*cf*. Fig. 2.6c), it becomes a correspondence matching problem with given 3D (sparse) points. In the following, we review the literature regarding each input data type.

Stereo-based scene flow. The most common setup is to jointly estimate 3D scene structure and 3D motion of each point given two temporally consecutive *stereo* images. Early approaches were mostly based on standard variational formulations and energy minimization, which defines an objective function that jointly estimates the disparity map at each time step and the 3D scene flow between the frames. From the estimated disparity, 3D point of each pixel in the world coordinate can be simply obtained from the baseline distance of a stereo rig and the camera intrinsics.

Early works followed the variational formulation or the standard pairwise MRF (Boykov et al., 1998; Li, 1994) formulation for optical flow based on the brightness constancy assumption as in Eqs. (2.1) and (2.2). Isard and MacCormick (2006) proposed a single MRF model that estimates the two disparity maps and the motion between them, considering occlusions, depth discontinuities, and motion discontinuities as well. Huguet and Devernay (2007) and Wedel et al. (2008) proposed a variational formulation that jointly estimates two disparity maps and scene flow by minimizing a defined global energy. However, these classical energy-based methods (Huguet and Devernay, 2007; Isard and MacCormick, 2006; Wedel et al., 2011; 2008; Zhang and Kambhamettu, 2001) often incur long runtime due to the energy optimization and yield limited accuracy particularly on occlusion, brightness changes, and non-texture region.

Later, Vogel et al. (2014, 2013b, 2015) introduced an explicit piecewise planar surface representation with a rigid motion model, and brought significant accuracy improvements especially in traffic scenarios. Vogel et al. (2013b) decompose a scene into a set of superpixels and estimate plane parameters and rigid motion parameters

(*i. e.*, rotation and translation in 3D space) of each superpixel. Further, Vogel et al. (2013b, 2015) extended the approach into a multiple frame setup that encouraged temporally consistency over time. Exploiting semantic knowledge for modeling motion of rigidly moving objects yielded further accuracy boosts (Behl et al., 2017; Ma et al., 2019; Menze and Geiger, 2015; Ren et al., 2017b).

Recently, CNN-based models have been introduced as well. Supervised approaches (Ilg et al., 2018; Jiang et al., 2019; Mayer et al., 2016; Saxena et al., 2019) train networks on large synthetic datasets and achieve state-of-the-art accuracy with real-time performance; yet, the domain mismatch problem toward real-world data still exists. To overcome the problem, un-/self-supervised learning approaches (Lee et al., 2019; Liu et al., 2019a; Wang et al., 2019b) had been also developed and were able to train their networks on real-world datasets directly.

RGB-D based scene flow. Another line of work (Hadfield and Bowden, 2011; Hornáček et al., 2014; Lv et al., 2018; Qiao et al., 2018; Quiroga et al., 2014) demonstrated estimating scene flow from a sequence of RGB-D images. As a depth measurement is provided by RGB-D sensor, the objective then becomes a correspondence matching problem in the 3D coordinate. Inspired by the classical optical flow formulation, Quiroga et al. (2014) proposed an energy-based model consisting of a data term and a pairwise term that encourages local and piecewise rigid motion in 3D space, followed by energy minimization. Hornáček et al. (2014) introduced an energy-based model that formulates the problem as local 3D patch matching under 6-DoF rigid motion.

Similar to other categories, CNN-based methods have been also developed. Qiao et al. (2018) proposed a CNN architecture that inputs a sequence of RGB-D images and directly regresses the scene flow. Lv et al. (2018) introduced a more structured CNN pipeline that estimates relative camera motion with a moving object mask as well as the scene flow.

Scene flow from 3D points cloud. Recently, several approaches explored to estimate scene flow given a sequence of 3D point clouds by using CNNs. Similar to RGB-D images, 3D point clouds also provide accurate 3D measurement of the surrounding environment.

Early CNN-based approaches proposed an end-to-end architecture that processes unstructured point clouds and directly regresses 3D scene flow. Due to the lack of ground truth for real data, the approaches train their networks on synthetic datasets and demonstrated generalization to real data. Gu et al. (2019) proposed a supervised learning approach based on a permutohedral lattice network that efficiently processes general, unstructured point-cloud data. Behl et al. (2019) introduced a voxel representation for handling sparse point cloud data and jointly estimate scene flow, ego-motion, and 3D bounding boxes for rigidly moving objects. Liu et al. (2019d) and Wang et al. (2020) demonstrated using set convolution, which takes neighboring point clouds within a certain radius and transforms them into a feature vector.

Afterward, self-supervised approaches demonstrated directly training on unlabeled point clouds. Mittal et al. (2020) proposed to use the cycle consistency between bi-directional scene flow and demonstrated comparable accuracy to supervised methods.

Further, Wu et al. (2020) introduced an improved architecture that follows a similar spirit on PWC-Net (Sun et al., 2018) (*i.e.*, using pyramids, warping, and cost volume) and outperformed the accuracy of previous methods.

Monocular scene flow. Scene flow estimation using a sequence of monocular images has been also proposed but less frequently. The main challenge comes from the depth and scale ambiguity, *i.e.* it is hard to determine them from a pair of consecutive frames.

Multi-task CNN approaches (Chen et al., 2019; Lai et al., 2019; Luo et al., 2019; Ranjan et al., 2019; Yang et al., 2018; Yin and Shi, 2018; Zhu et al., 2019; Zou et al., 2018b) that jointly predict optical flow, depth, and camera motion from a monocular sequence demonstrate reconstructing scene flow from those outputs. However, such approaches have a critical limitation in that they cannot recover scene flow for occluded pixels.

Xiao et al. (2017) introduced a variational approach to monocular scene flow given an initial depth cue, but without competitive accuracy. Brickwedde et al. (2019) proposed an integrated pipeline by combining CNNs and an energy-based formulation. Given depth estimates from monocular depth CNNs, trained on pseudo-labeled data, the method jointly estimates 3D plane parameters and the 6D rigid motion of a piecewise rigid scene representation, achieving state-of-the-art accuracy. Yang and Ramanan (2020) introduced an integrated pipeline that obtains scene flow from given optical flow and depth cues via determining motion in depth from observing changes in object sizes.

In Chapter 6, we propose a monocular scene flow approach that yields both competitive accuracy and real-time performance by directly predicting 3D scene flow from a single CNN. Due to the scarcity of 3D motion ground truth and the domain over-fitting problem when using synthetic datasets, we train directly on the target domain in a self-supervised manner, which can leverage large amounts of unlabeled data.

2.3 JOINT ESTIMATION WITH MOTION

As discussed in Chapter 1, the dissertation mainly focuses on the joint estimation of motion with *semantic segmentation, occlusion, and depth*. Here, we introduce the background and literature on each joint objective with motion. Especially, we discuss how the joint objectives have been formulated and what kinds of relationships have been exploited in previous work, along with our contributions.

2.3.1 *Semantic Segmenatation*

Semantic segmentation is the task of classifying each pixel in an image into one of pre-defined object class labels. As in Fig. 2.7, it assigns an object class label $l \in L$ to each pixel $\mathbf{p} = (p_x, p_y)$ in the image $\mathbf{I}_t$, where L is a set of semantic classes (*e.g.*, sky, building, road, car, pedestrian, etc.).

Figure 2.7: Examples of semantic segmentation (Ground truth from the Cityscapes dataset (Cordts et al., 2016)).

Prior to the deep learning age, classical approaches (Arbeláez et al., 2012; Kontschieder et al., 2011; Nowozin et al., 2010; Xia et al., 2013) used handcrafted low-level features (*e. g.* SIFT (Lowe, 2004) or HoG (Dalal and Triggs, 2005)), followed by graphical models (*e. g.* MRFs, Conditional Random Fields (CRFs)) or Random Forest. Yet, their lack of representation capability as well as ability to handle a large-scale data limits their performance.

Afterward, the relatively recent success of CNNs with back-propagation enables to train models on large-scale datasets. Continuing research on finding better CNN architectures (Chen et al., 2017b; Yu and Koltun, 2016; Zhao et al., 2017) as well as establishing diverse large scale datasets (Cordts et al., 2016; Neuhold et al., 2017) accelerate the progress of research.

While the majority of the work focuses on per-frame estimation, exploiting multiple frames at a single time step can bring new aspects to the problem, which is one of the main interests of our dissertation. One can simply exploit the temporal correspondence between frames to improve the temporal consistency of segmentation. On the other hand, one can also improve motion estimation by exploiting semantic knowledge. Or, one can improve both tasks by discovering mutual benefits between them. Below, we discuss the related prior works on each case.

Temporally consistent semantic segmentation. Given multiple temporally consecutive frames, temporal correspondence information between frames can improve the semantic segmentation accuracy by encouraging temporal consistent estimation between corresponding pixels. One common way to inject temporal consistency is to utilize motion and structure features from 3D point clouds obtained by Structure from Motion (SfM) (Brostow et al., 2008; Floros and Leibe, 2012; Sturgess et al., 2012). Another way is to jointly reconstruct a scene in 3D with semantic labels through a batch process, which naturally enables temporally consistent segmentation (Kundu et al., 2014; Sengupta et al., 2013; Zhang et al., 2010).

In causal approaches that rely on temporal correspondence, previous methods achieved accurate temporal correspondence using sparse feature tracking (Scharwächter et al., 2014) or dense flow maps with a similarity function in feature space (Miksik et al., 2013). Kundu et al. (2016) introduced feature space optimization for spatio-temporal regularization in partitioned batches with overlaps. With CNNs, several approaches (Gadde et al., 2017; Nilsson and Sminchisescu, 2018; Paul et al., 2020) demonstrated propagating a feature map from the previous time step by warping it using optical flow, and improved the temporal consistency between the estimation over multiple frames. However, because those methods (Gadde et al., 2017; Nilsson

and Sminchisescu, 2018; Paul et al., 2020) use off-the-shelf optical flow methods, their accuracy depends on the accuracy of the off-the-shelf methods.

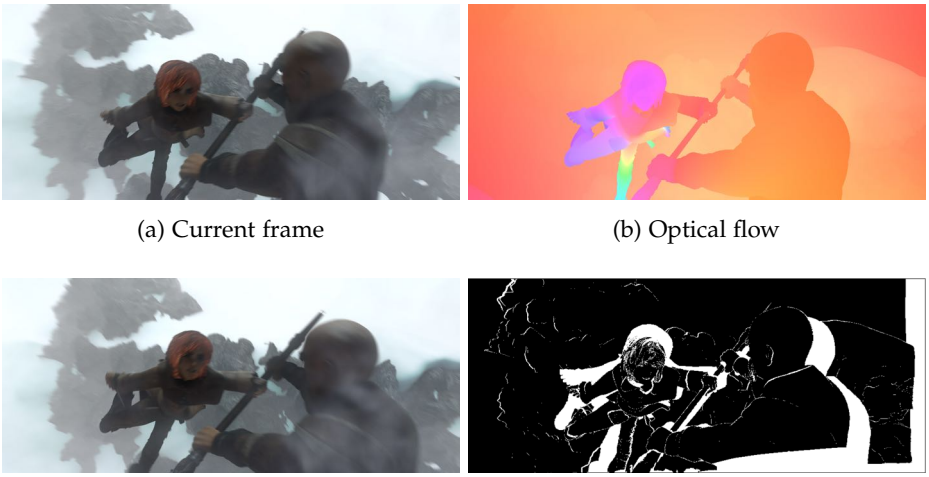
Motion estimation with semantics. The other unidirectional way has been explored through exploiting semantic knowledge for motion estimation. Sevilla-Lara et al. (2016) proposed a sequential approach for optical flow: the method first segments the scene into three semantic categories (*e. g.* things, planes, and stuff), estimates motion of these semantic parts individually, and then composes them in the end. For scene flow, a couple of approaches exploits (Behl et al., 2017; Ren et al., 2017b) moving object cues to separately model motion of moving objects and background, and demonstrate better accuracy.

Joint semantic segmentation and optical flow estimation. Cheng et al. (2017) demonstrated joint video object segmentation and optical flow estimation using two separate CNNs with skip connections in the decoder part. Through a joint training, Cheng et al. (2017) demonstrated improving the accuracy of both segmentation and optical flow. Jiang et al. (2019) introduced a shared encoder CNN model that outputs disparity, optical flow, and semantic segmentation through each decoder. Jiang et al. (2019) showed that their joint model performs on par with separate models for each task but sometimes with marginal improvement. Ding et al. (2020) introduced a CNN model that jointly estimates video semantic segmentation and optical flow. Differently from Cheng et al. (2017), Ding et al. (2020) include occlusion reasoning so that the estimates on occluded areas do not affect the next frame, which increases the accuracy of both tasks.

In Chapter 3, we also demonstrate the joint estimation of the two tasks and overcome the limitation of previous unidirectional methods (Gadde et al., 2017; Nilsson and Sminchisescu, 2018; Paul et al., 2020; Sevilla-Lara et al., 2016), in which their accuracy is bounded by off-the-shelf methods they use. In contrast to previous work on joint estimation (Cheng et al., 2017; Ding et al., 2020; Jiang et al., 2019), our approach explicitly designs the motion model depending on its semantic information, for instance, by constraining that static objects should follow relative camera motion. This is different from the CNN-based methods (Cheng et al., 2017; Ding et al., 2020; Jiang et al., 2019) that implicitly learn from data and thus lack explainability.

2.3.2 *Occlusion*

Occlusion is one of the phenomena that is observed in the 2D image coordinate when motion exists in 3D space. Given a 3D scene with non-transparent objects, moving objects or an ego-motion of the camera blocks the sight of other objects that are no longer visible in the next frame. The areas, in the current image, that cannot be observed in the next image are considered as occlusion. The areas that go out of the image boundary and disappear in the next image are called Out-of-Bound (OoB) pixels, which is also considered as occlusion. Fig. 2.8b and Fig. 2.8d visualize the optical flow and the corresponding occlusion between the current frame (Fig. 2.8a) and the next frame (Fig. 2.8c), as an example.



(a) Current frame (b) Optical flow

(c) Next frame (d) Occlusion

Figure 2.8: **Visualization of optical flow and occlusion:** Visualization of the ground truth of *(b)* optical flow and *(d)* occlusion between *(a)* the current frame and *(c)* the next frame. Images are from MPI Sintel dataset (Butler et al., 2012).

Occlusion itself rather serves as supplementary information for other computer vision objectives. For instance, in correspondence matching tasks such as optical flow, descriptor matching, or semantic matching, accurate occlusion information provides which areas are not reliable or do not need to attend. Occlusion also provides a depth ordering cue in the 2D image space, and this can be informative for the video interpolation task where foreground and background pixels need to be distinguished.

In the context of optical flow estimation, occlusion estimation is a well-known chicken-and-egg problem that optical flow has been entangled with for a long time (Alvarez et al., 2007; Kolmogorov and Zabih, 2001; Pérez-Rúa et al., 2016; Strecha et al., 2004). Accurate knowledge of occluded areas is crucial for reliable optical flow estimation in order to prevent non-occluded areas from being adversely affected by occluded pixels. Yet, occlusion is a consequence of motion: accurate optical flow estimation, conversely, is required for localizing occlusions reliably. In the following, we briefly discuss how the previous literature has formulated the relationship between the two tasks, and then connect with our contributions.

Occlusions as outliers. As discussed above, the chicken-and-egg problem of flow and occlusion estimation becomes much simpler if occlusions are treated as outliers that violate the basic optical flow assumptions (*e. g.*, brightness constancy and/or forward-backward flow consistency assumptions). A number of recent methods (Bailer et al., 2015; Chen and Koltun, 2016; Gadot and Wolf, 2016; Güney and Geiger, 2016; Hu et al., 2016; Li et al., 2015; Menze et al., 2015a) follow a common strategy for outlier filtering. Based on a robust, truncated data term, they *(i)* separately estimate forward and backward flow with an asymmetric method, *(ii)* conduct a bi-directional consistency check, and *(iii)* interpolate flow into the outlier pixels in a post-processing stage. With the aid of highly capable interpolation methods (Li et al., 2016; Revaud et al., 2015), this pipeline has been regarded as a well-justified practice. However, occasional

failures during post-processing are inevitable and irreversible, and thus constitute a fundamental limitation.

In the supervised learning setup using CNNs, Zhao et al. (2020) demonstrated an asymmetric occlusion-aware feature matching module that learns to estimate a rough occlusion mask without explicit supervision and filters out the warped feature map in the occluded area so that the doubled feature map does not affect the matching process. In an un-/self-supervised way using CNNs, a series of works detect occlusion (Meister et al., 2018; Wang et al., 2018) or estimate occlusion using a decoder (Janai et al., 2018) and discard the occluded area when calculating their proxy loss functions.

Occlusions in a joint objective. An outlier is a failure of flow estimation, but an occlusion is a consequence of motion, and can conversely be used as additional evidence for estimating optical flow. Distinguishing between the two opens new possibilities. A number of previous works consider occlusion explicitly in the formulation but received less attention. We aim to bring them back into focus, revisit their ideas, and highlight the importance of jointly handling occlusions as a feature complementary to other recent trends, including the use of deep learning for appearance matching (Bai et al., 2016; Dosovitskiy et al., 2015; Güney and Geiger, 2016; Ilg et al., 2017).

All methods in this category begin with including an occlusion variable in their objective. Yet, they differ in the particular characteristics of occlusion utilized and how these are formulated. One basic way to characterize occlusion stems from the observation that *brightness constancy* mostly does not hold in occlusion areas due to the non-existence of corresponding pixels. Several approaches (Sun et al., 2014a; Unger et al., 2012; Xiao et al., 2006) adopt a constant penalty (or truncated cost) in the data term so that it can *(i)* naturally lead to occluded pixels taking the constant penalty rather than a potentially higher matching cost, and *(ii)* explicitly exclude their matching cost from the objective (*e.g.*, as visualized in Fig. (6) of Sun et al. (2014a)). However, this property alone is not sufficient (Xiao et al., 2006) as it is impossible to discriminate between occluded pixels and pixels with strong illumination changes, which both violate the assumption.

Introducing a *forward-backward consistency constraint* into the objective function is another useful strategy. Its benefit is that pixels are forced to be either visible and satisfy the bi-directional flow consistency, or are identified as occlusions (Ince and Konrad, 2008). This condition provides an additional cue for joint estimation. Yet, one should not forget that occlusion is a consequence of two different motions causing one pixel to geometrically occlude another. Layered optical flow models (Sun et al., 2014a; 2010b) or 3D scene flow methods (Vogel et al., 2014) can explicitly model the local depth relationship between layers and estimate occlusions simultaneously. Similarly, one can calculate overlapping areas between two triangular patches and detect occlusions by comparing the photometric cost (Kennedy and Taylor, 2015). Calculating the divergence of the motion field (Ballester et al., 2012) or finding unique configurations of corresponding pixels (Kolmogorov and Zabih, 2001; Unger et al., 2012) can be alternative approaches. Alvarez et al. (2007) exploited symmetry properties for jointly estimating optical flow and occlusion by encouraging forward-backward flow consistency between corresponding pixels.

More recently using CNNs, Ilg et al. (2018) proposed a supervised learning approach for jointly estimating optical flow and occlusion, as well as depth and motion boundaries.

In Chapter 4, we demonstrate a symmetric approach that explicitly integrates occlusions into the objective in order to exploit them as an important cue for the flow itself. Unlike the approaches considering occlusions as outliers (Bailer et al., 2015; Chen and Koltun, 2016; Gadot and Wolf, 2016; Güney and Geiger, 2016; Hu et al., 2016; Li et al., 2015; Menze et al., 2015a), our integrative approach simultaneously estimates flow in both directions and thus encourages bi-directional consistency of the flow as part of the formulation, which naturally makes any post-processing unnecessary. Furthermore, our model exploits the relations and symmetry properties between optical flow and occlusion jointly and more completely than previous works, leading to significant benefits in accuracy.

In Chapter 5, we also demonstrate a CNN architecture that jointly estimates optical flow and occlusion. Our method is a supervised-learning approach similar to Ilg et al. (2018), but demonstrates improving the flow accuracy with occlusion estimation, unlike Ilg et al. (2018).

2.3.3 *Depth*

In this dissertation, we are interested in estimating depth using two temporally consecutive frames. Unlike the stereo matching problem using rectified stereo images with a given baseline distance, we aim to estimate depth in a more general but challenging case where two frames are taken from the same (moving) camera, with unknown camera ego-motion.

Recently, a number of Deep Neural Network (DNN)-based approaches have demonstrated depth estimation using multiple images or even a single monocular image (Chen et al., 2019; Lai et al., 2019; Ranjan et al., 2019; Yang et al., 2018; Yin and Shi, 2018; Zhu et al., 2019; Zou et al., 2018b). The key idea is to learn scene prior knowledge from training samples, sometimes jointly estimating with motion, ego-motion of the camera, and knowledge of moving objects. Here, we mainly discuss the work that jointly estimates depth with motion.

Joint estimation of optical flow and depth. Along with the recent advances in CNNs, a couple of works (Ummenhofer et al., 2017; Zhou et al., 2018) demonstrated jointly estimating depth and ego-motion by exploiting motion parallax in a supervised learning setup. However, relatively poor generalization to real world images or not being able to handle moving objects limits their applicability towards general, dynamic scenes.

In contrast, a number of methods (Chen et al., 2019; Lai et al., 2019; Ranjan et al., 2019; Yang et al., 2018; Yin and Shi, 2018; Zhu et al., 2019; Zou et al., 2018b) recently demonstrated joint estimation of optical flow and depth by training CNNs in an unsupervised or self-supervised manner. Though their accuracy is not competitive to the supervised-learning approaches yet, it drew attention that it is possible to jointly estimate multiple tasks (*e. g.* depth, motion, camera motion, or moving object mask)

by directly training in the target domain without annotated data. The key technical idea is to exploit the view synthesis as a proxy task: one trains the network to output motion, depth, and/or ego-motion in order to properly synthesize the reference view from the next view, based on two-view geometry.

In Chapter 6, we demonstrate estimating depth and 3D scene flow using CNNs with a single, joint decoder. Previous multi-task approaches (Luo et al., 2019; Yang et al., 2018) demonstrated estimating depth and 2D optical flow, which possibly reconstructs scene flow from their outputs; yet, these methods yield limited scene flow accuracy due to being limited to non-occluded regions. In contrast, our method directly estimates 3D scene flow with a CNN so that we naturally bypass this problem. Besides, while these multi-task CNN methods often require complex training schedules to balance between multiple decoders, our method demonstrates using a simple, practical training schedule by virtue of using a single joint decoder.

3

JOINT OPTICAL FLOW AND TEMPORALLY CONSISTENT SEMANTIC SEGMENTATION

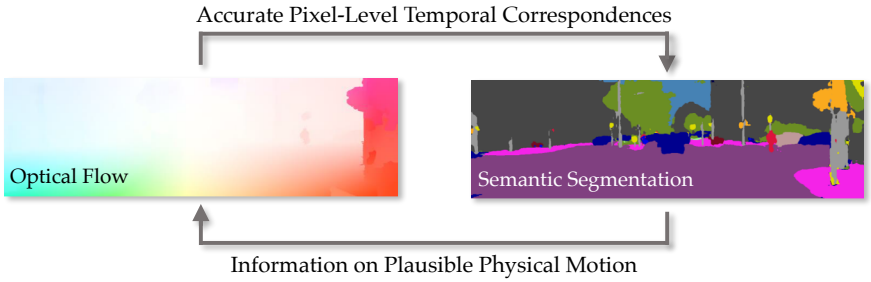


Figure 3.1: **Our joint objective**: Optical flow provides accurate pixel-level temporal correspondences to semantic segmentation, and semantic knowledge provides information on physically plausible motion for optical flow estimation.

CONTENTS

As discussed in Chapter 1, we explore our first joint objective in this chapter: optical flow and semantic segmentation. Given two temporally consecutive frames, we propose a method for the joint estimation of optical flow and temporally consistent semantic segmentation, which closely connects these two problem domains and allows each task leverage the other. Semantic segmentation provides information on plausible physical motion to its associated pixels, and accurate pixel-level temporal

correspondences enhance the accuracy of semantic segmentation in the temporal domain. We demonstrate the benefits of our approach on the KITTI benchmark, where we observe performance gains for both flow and segmentation. Our method substantially outperforms direct competitors on challenging, but crucial dynamic objects.

3.1 INTRODUCTION

Visual scene understanding from movable platforms has gained increasing attention in part due to the active development of autonomous systems and vehicles. Semantic segmentation and dense motion estimation are two core components of dynamic scene understading. Attributable to the establishment of public benchmarks (*e.g.*, KITTI (Geiger et al., 2012; Menze et al., 2015b; 2018), MPI Sintel (Butler et al., 2012), or Cityscapes (Cordts et al., 2016)), the performance of techniques, popularity and importance of both areas have been increasing and well acknowledged. In line with this, a growing body of literature focuses on how to bridge the two themes efficiently and discover additional benefits that each task can obtain from the other.

There have been a few preliminary attempts to utilize optical flow to enforce temporal consistency of semantic segmentation in a video sequence (Chen and Corso, 2011; Grundmann et al., 2010; Miksik et al., 2013). Also, segmention of the scene into superpixels (without clear semantics) has been shown to help estimate more accurate optical flow, assuming that object boundaries may lead to motion boundaries (Sun et al., 2014a; Yamaguchi et al., 2013; 2014). Strictly speaking, however, previous work has simply used the results from one task to supplement the other, and attempts to investigate a close relationship between the two tasks remain lacking. The accuracy of off-the-shelf motion estimation algorithms is not settled yet (Chen and Corso, 2011; Miksik et al., 2013). The only exception is the work by Sevilla-Lara et al. (2016), which uses both semantic information and segmentation to increase the accuracy of optical flow; however, it did not consider the benefits of temporal correspondence for semantic labeling. Please refer to Section 2.3.1 for a more comprehensive literature review.

In this chapter, we address this gap and present an approach for joint optical flow estimation and temporally consistent semantic segmentation from a monocular video, in which each task leverages the other. Fig. 3.2 shows the overview of our method. We begin with the assumption that a bottom-up semantic segmentation for each frame is given. Then we estimate accurate optical flow fields by exploiting the semantic information from the given semantic segmentation. The benefit of having semantic labels is that they can yield information on the likely physical motion of the associated pixels. At the same time, accurate pixel-level correspondence between consecutive frames can establish temporally consistent semantic segmentation and help refine the initial results.

We make two major contributions in this chapter. First, we introduce an accurate piecewise parametric optical flow formulation, which itself already outperforms the state of the arts, particularly on dynamic objects. Our formulation explicitly

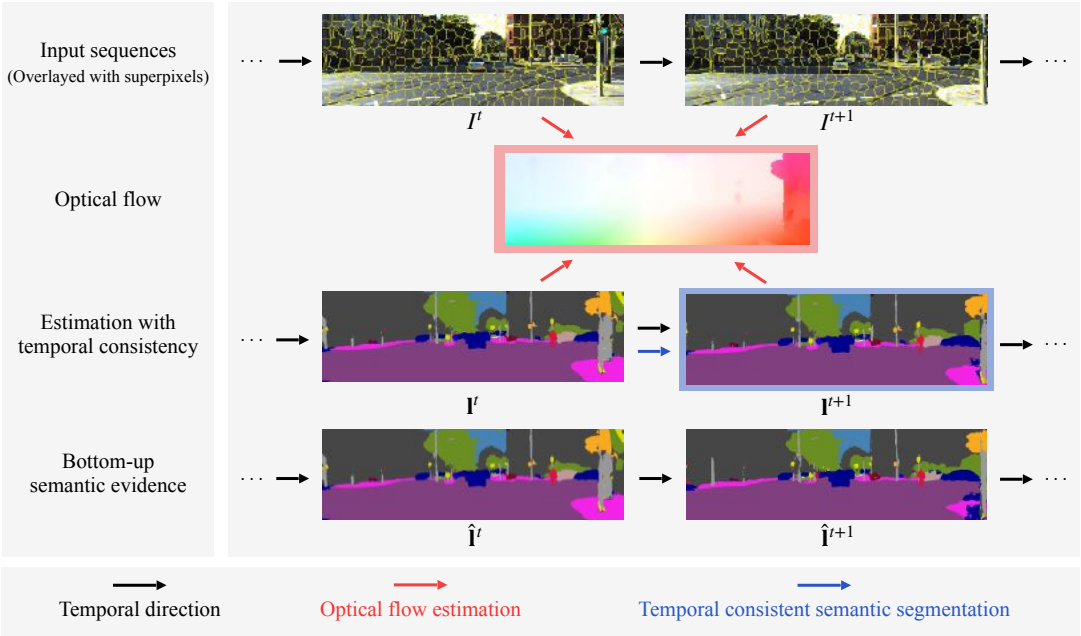


Figure 3.2: **Overview of our approach.** The red arrows contribute to optical flow estimation, and the blue arrows ensure temporal consistency of the semantic segmentation, both given two frames. Input images are overlayed with superpixels. The images highlighted in color defines the output of our method.

handles occlusions to prevent the data term from excessively influencing the results in occlusion areas. As a result, our method additionally provides occlusion information such as occlusion masks and occlusion types. Our second contribution is the joint estimation of optical flow and temporally consistent semantic segmentation in a monocular video setting. For the flow estimation, we additionally apply the epipolar constraint for pixels that should be consistent with the camera ego-motion, as inferred by the semantic information. At the same time, accurately estimated flow helps enforce temporal consistency on the semantic segmentation. We effectively conceptualize these ideas in our joint formulation and make them feasible using inference based on PatchMatch Belief Propagation (PMBP) (Besse et al., 2013).

Our experiments on the popular KITTI dataset show that our method yields state-of-the-art results for optical flow. For estimating flows on dynamic foreground objects, which are particularly crucial for autonomous navigation, our method substantially outperforms all published optical flow algorithms in the benchmark at the time of the benchmark submission.

3.2 APPROACH

The core idea put forward in this chapter is that optical flow and semantic segmentation are mutually beneficial and are best estimated jointly to simultaneously improve each other. Fig. 3.2 shows the flow of our proposed method in the temporal domain and explains which elements contribute to achieving which task. Here, we assume

that some initial bottom-up semantic evidence is already given by an off-the-shelf algorithm, such as a CNN (*e. g.*, Long et al. (2015)), which is subsequently refined by having temporal consistency. As highlighted in red, a pair of consecutive images and their refined semantic segmentation contribute to estimating optical flow more accurately. At the same time, as highlighted in blue, the temporally consistent semantic labeling at time $t+1$ is inferred from its bottom-up evidence, the previously estimated semantic labeling at time t, and the estimated flow map. For longer sequences, our approach proceeds in an online manner on two frames at a time.

Similar to Yang and Li (2015), our formulation is based on an 8-DoF piecewise-parametric model with a superpixelization of the scene. Superpixels play an important role in our formulation for connecting the two different domains: optical flow and semantic segmentation. One superpixel represents a global motion as well as a semantic label for its pixels inside, and the motion is constrained by the physical properties that the semantic label implies. For example, the motion of pixels corresponding to some physically-static objects (*e. g.*, building or road) can only be caused by camera motion. Thus, enforcing the epipolar constraint on those pixels can effectively regularize their motion. This is different from previous works that rely on the epipolar constraint for estimating motion, which demonstrate an inherent limitation for handling independently moving objects whose motion usually violates the constraint (Kitt and Lategahn, 2012; Mohamed et al., 2015; Yamaguchi et al., 2013; 2014).

Another important feature of our formulation is that we explicitly formulate the occlusion relationship between superpixels (Yamaguchi et al., 2013; 2014) and infer the occlusion mask as well. This directly affects the data term such that it prevents occluded pixels from dominating the data term during the optimization.

3.2.1 *Preprocessing*

Superpixels. As superpixels generally tend to separate objects in images, they can be a good medium for carrying semantic labels and representative motions for their pixels. Our approach uses the recent state-of-the-art work of Yao et al. (2015), which has shown to be well suited for estimating optical flow.

Semantic segmentation. For the bottom-up semantic evidence, we use an off-the-shelf Fully Convolutional Network (FCN) (Long et al., 2015) trained on the Cityscapes dataset (Cordts et al., 2016), which contains typical objects frequent in street scenes.

Fundamental matrix estimation. In order to apply the epipolar constraint on superpixels for which their semantic label tells us that they are surely static objects (*e. g.*, roads, buildings, etc.), our approach requires the fundamental matrix resulting from the camera motion. We use a standard approach, *i. e.* matching SIFT keypoints (Lowe, 2004) and using the 8-point algorithm (Hartley, 1997) with RANSAC (Lourakis, 2011).

3.2.2 *Model*

Our model jointly estimates *(i)* the optical flow between reference frame I^t and the next frame I^{t+1}, and *(ii)* a temporally consistent semantic segmentation $\mathbf{l}^{t+1}$ given bottom-up semantic evidence $\hat{\mathbf{l}}^{t+1}$ and the previously estimated semantic labeling $\mathbf{l}^t$. $\mathbf{l}$ is a semantic label probability map, which has the same size as the input image and L channels, where L is the number of semantic classes. Instead of using a single label, we adopt label probabilities so that we can more naturally and continuously infer the semantic labels in the time domain. Note that we assume an online setting (*i. e.*, no access to future information) and hence infer the segmentation at time $t+1$ rather than t. Optical flow is represented by a set of piecewise motions of superpixels in the reference frame. We define the motion of a superpixel through a homography and formulate the objective for estimating the 8-DoF homography $\mathbf{H}_s$ of each superpixel s and the temporally consistent semantic segmentation $\mathbf{l}^{t+1}$ as:

$$\begin{aligned} E(\mathbf{H}, \mathbf{l}^{t+1}, o, b) = E_{\mathrm{D}}(\mathbf{H}, o) + \lambda_{\mathrm{L}} E_{\mathrm{L}}(\mathbf{H}, \mathbf{l}^{t+1}, o) \\ + \lambda_{\mathrm{P}} E_{\mathrm{P}}(\mathbf{H}) + \lambda_{\mathrm{C}} E_{\mathrm{C}}(\mathbf{H}, o, b) + \lambda_{\mathrm{B}} E_{\mathrm{B}}(b), \end{aligned} \tag{3.1}$$

consisting of color data term E_{D}, label data term E_{L}, physical constraint term E_{P}, connectivity term E_{C}, and boundary occlusion prior term E_{B}.

We adopt two kinds of occlusion variables: the boundary occlusion label b between two superpixels, and the occlusion mask o defined at the pixel level. The boundary occlusion label b regularizes the spatial relationship between two neighboring superpixels (*i. e.*, co-planar, hinge, left occlusion, or right occlusion) (Yamaguchi et al., 2013; 2014). The occlusion mask o explicitly models whether a pixel is occluded or not. One important difference to previous superpixel-based work (Yamaguchi et al., 2014) is that we additionally infer a pixelwise occlusion mask, which prevents occluded pixels from adversely affecting the data cost.

Data terms. The data terms aggregate photometric differences

$$E_{\mathrm{D}}(\mathbf{H}, o) = \sum_{s \in S} \frac{1}{|s|} \underbrace{\sum_{\mathbf{p} \in s} (1 - o_{\mathbf{p}}) \rho_{\mathrm{D}} \left(I^t(\mathbf{p}), I^{t+1}(\mathbf{p}') \right) + o_{\mathbf{p}} \lambda_{\mathrm{o}}}_{\text{image data}} \tag{3.2}$$

and semantic label differences

$$E_{\mathrm{L}}(\mathbf{H}, \mathbf{l}^{t+1}, o) = \sum_{s \in S} \frac{1}{|s|} \sum_{\mathbf{p} \in s} \phi_l(\mathbf{H}, \mathbf{l}^{t+1}_{\mathbf{p}'}, o) \tag{3.3a}$$

where

$$\phi_l(\mathbf{H}, \mathbf{l}^{t+1}_{\mathbf{p}'}, o) = \frac{1}{2} \sum_{i}^{L} (1 - o_{\mathbf{p}}) \left\| \mathbf{l}^{t+1}_{\mathbf{p}',i} - (\alpha \hat{\mathbf{l}}^{t+1}_{\mathbf{p}',i} + (1 - \alpha) \mathbf{l}^{t}_{\mathbf{p},i}) \right\|^2 \tag{3.3b}$$

over each pixel of each superpixel. Here, $\mathbf{p}'$ is the corresponding pixel in I^{t+1} of pixel $\mathbf{p}$ in I^t, which is determined according to the homography $\mathbf{H}_{S(\mathbf{p})} \in \mathbb{R}^{3 \times 3}$ of its superpixel:

$$\mathbf{p}' = \mathbf{H}_{S(\mathbf{p})} \mathbf{p}, \tag{3.4}$$

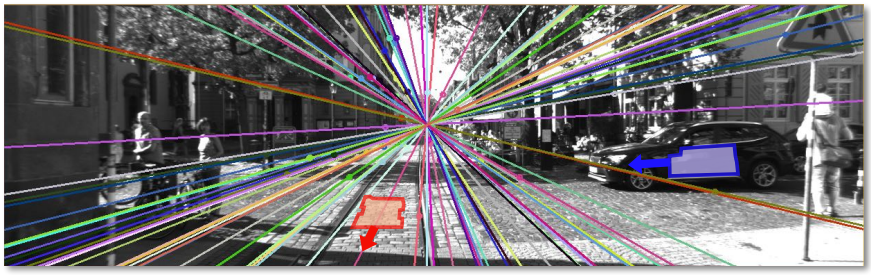


Figure 3.3: **Physical constraint term** is only applied on the (super)pixel belonging to physically static objects which 2D motion should follow the epipolar motion (*e. g.*, the red superpixel on the road *vs.* the blue superpixel on the vehicle).

where $\mathcal{S} : I^t \rightarrow S$ is a mapping that assigns a pixel $\mathbf{p}$ to its superpixel $s \in S$.

In the image data term in Eq. (3.2), the function $\rho_D(\cdot, \cdot)$ measures the photometric differences between two pixels using the ternary transform (Stein, 2004) and a truncated linear penalty. If a pixel $\mathbf{p}$ is occluded (*i. e.*, $o_\mathbf{p} = 1$), a constant penalty λ_o is applied.

The label data term in Eq. (3.3a) measures the distance between two semantic label probability distributions over each pixel: *(i)* our estimation $\mathbf{l}_{\mathbf{p}'}^{t+1}$ and *(ii)* a weighted sum of the previous estimation $\mathbf{l}_\mathbf{p}^t$, which is propagated by the optical flow, and the bottom-up evidence $\hat{\mathbf{l}}_{\mathbf{p}'}^{t+1}$, while considering its occlusion status. The motivation of the term is to penalize label differences to the bottom-up evidence and at the same time propagate label evidence over time, except when an occlusion takes place.

Physical constraint term. Semantic labels can provide useful cues for estimating optical flow. If pixels are labeled as physically static objects, such as building, road, or infrastructure (*e. g.*, the red-colored superpixel in Section 3.2.2), then they normally do not undergo any 3D motion, hence their observed 2D motion is caused by camera motion and should thus satisfy the epipolar constraint. We define the corresponding term as

$$E_\mathrm{P}(\mathbf{H}) = \sum_{s \in S} \min(\phi_P(s, \mathbf{H}_s), \lambda_{\text{non\_st}} + \beta[l_s^t \in L_{\text{st}}]), \tag{3.5}$$

where

$$\phi_{\text{st}}(s, \mathbf{H}_s) = \frac{1}{|s|} \sum_{\mathbf{p} \in s} \left\| \mathbf{p}'^\top \mathbf{F} \mathbf{p} \right\|_1 = \frac{1}{|s|} \sum_{\mathbf{p} \in s} \left\| (\mathbf{H}_{\mathcal{S}(\mathbf{p})} \mathbf{p})^\top \mathbf{F} \mathbf{p} \right\|_1 \tag{3.6}$$

measures how well the homography matrix $\mathbf{H}_s$ of a superpixel s meets the epipolar constraint from the fundamental matrix $\mathbf{F}$. For non-static objects, such as pedestrians or vehicles (*e. g.*, the blue-colored superpixel in Section 3.2.2), we still apply the epipolar penalty, however a weak one using a low truncation threshold $\lambda_{\text{non\_st}}$. This is motivated by the fact that possibly dynamic objects may in fact stand still and thus obey epipolar geometry, but we do not want to penalize them too much if they do not. For static objects, on the other hand, we augment the truncation threshold by β in order to give a stricter penalty. L_{st} is the set of semantic labels that corresponds to the

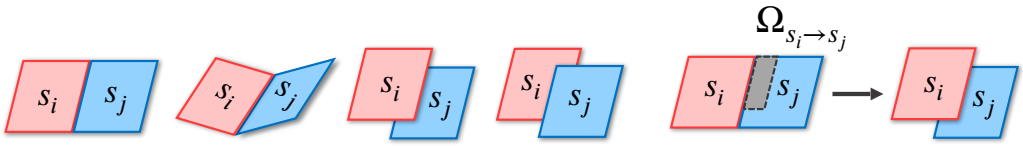


Figure 3.4: **Boundary relatinship between superpixels**: *(a)* Four cases of boundary relations between two superpixels: co-planar, hinge, left occlusion, and right occlusion. *(b)* The visualization of the set of occluded pixels $\Omega_{s_i \to s_j}$ in the case of a left occlusion (black-colored region).

physically static objects. l_s is a representative semantic label of superpixel s, which has the highest probability over the pixels in the superpixel: $l_s = \text{argmax}_i \sum_{\mathbf{p} \in s} \mathbf{l}^t_{\mathbf{p},i}$.

Connectivity term. The connectivity term encourages the smoothness of motion between two neighboring superpixels based on their occlusion relationship:

$$E_{\text{C}}(\mathbf{H}, o, b) = \sum_{s_i \sim s_j} \phi_{\text{C}}(\mathbf{H}_{s_i}, \mathbf{H}_{s_j}, o, b_{ij}) \tag{3.7}$$

with

$$\phi_{\text{C}}(\mathbf{H}_{s_i}, \mathbf{H}_{s_j}, o, b_{ij}) = \begin{cases} \phi_{\text{co}}(\mathbf{H}_{s_i}, \mathbf{H}_{s_j}, o) & \text{if } b_{ij} = \text{co-planar}, \\ \phi_{\text{h}}(\mathbf{H}_{s_i}, \mathbf{H}_{s_j}, o) & \text{if } b_{ij} = \text{hinge}, \\ \phi_{\text{occ}}(s_i, s_j, o) & \text{if } b_{ij} = \text{left occlusion}, \\ \phi_{\text{occ}}(s_j, s_i, o) & \text{if } b_{ij} = \text{right occlusion}. \end{cases} \tag{3.8}$$

As shown in Fig. 3.4a, the boundary occlusion flag b_{ij} expresses the relationship between two neighboring superpixels s_i and s_j as co-planar, hinge, left-occlusion, or right-occlusion (Yamaguchi et al., 2013; 2014). This categorization helps to regularize the motion of two superpixels defined by their homography matrices. We distinguish between three different potentials:

$$\phi_{\text{co}}(\mathbf{H}_{s_i}, \mathbf{H}_{s_j}, o) = \frac{1}{|s_i \cup s_j|} \sum_{\mathbf{p} \in s_i \cup s_j} \|\mathbf{H}_{s_i}\mathbf{p} - \mathbf{H}_{s_j}\mathbf{p}\|_1 + \sum_{\mathbf{p} \in s_i \cup s_j} \lambda_{\text{imp}}[o_p = 1] \tag{3.9a}$$

$$\phi_{\text{h}}(\mathbf{H}_{s_i}, \mathbf{H}_{s_j}, o) = \frac{1}{|\mathcal{B}_{s_i,s_j}|} \sum_{\mathbf{p} \in \mathcal{B}_{s_i,s_j}} \|\mathbf{H}_{s_i}\mathbf{p} - \mathbf{H}_{s_j}\mathbf{p}\|_1 + \sum_{\mathbf{p} \in s_i \cup s_j} \lambda_{\text{imp}}[o_p = 1] \tag{3.9b}$$

$$\begin{aligned} \phi_{\text{occ}}(s_{\text{f}}, s_{\text{b}}, o) = \sum_{\mathbf{p} \in s_{\text{b}}} \Big(& \lambda_{\text{imp}}[\mathbf{p} \in \Omega_{s_{\text{f}} \to s_{\text{b}}}][o_p = 0] \\ & + \lambda_{\text{imp}}[\mathbf{p} \notin \Omega_{s_{\text{f}} \to s_{\text{b}}}][o_p = 1] \Big) + \sum_{\mathbf{p} \in s_{\text{f}}} \lambda_{\text{imp}}[o_p = 1] \end{aligned} \tag{3.9c}$$

These are motivated as follows: When two superpixels are co-planar, all pixels within should follow the identical homography matrix as they are on the same plane. For a hinge relationship, only the pixels on the boundary set $\mathcal{B}_{s_i,s_j}$ can satisfy the motion from two superpixels s_i and s_j. In both cases, there should be no occluded pixels, hence we adopt a very large 'impossible' penalty λ_{imp} to prevent occluded pixels

from occurring. In case that one superpixel occludes another, their motions only affect the occlusion masks. Eq. (3.9c) expresses the case that pixels of the front superpixel s_f occlude some pixels of the back superpixel s_b. As shown in Fig. 3.4b, $\Omega_{s_f \to s_b}$ is a set of pixels in s_b that is occluded by some pixels in s_f from the motion. All pixels in the front superpixel s_f should not be occluded, and only pixels in the set of $\Omega_{s_f \to s_b}$ in s_b should be occluded.

Boundary occlusion prior. Without an additional prior term, the boundary occlusion flag in the connectivity term would prefer to take the occlusion cases. We thus define a prior term to yield proper biases for each case:

$$E_B(b) = \begin{cases} \lambda_{co}[l_{s_i} \neq l_{s_j}] & \text{if } b_{ij} = \text{co-planar}, \\ \lambda_h & \text{if } b_{ij} = \text{hinge}, \\ \lambda_{occ} & \text{if } b_{ij} = \text{occlusion}, \end{cases} \tag{3.10}$$

where $\lambda_{occ} > \lambda_h > \lambda_{co} > 0$. Because it is less likely that two different objects are co-planar in the real world, we only apply the prior penalty for the co-planar case λ_{co} when the respective semantic labels of the superpixels differ.

3.2.3 *Optimization*

The minimization of our objective is challenging, as it combines discrete (*i. e.*, $\{l^{t+1}, b, o\}$) and continuous (*i. e.*, $\mathbf{H}$) variables. We use a block coordinate descent algorithm. As shown in Alg. 1, we iteratively update each variable in the order: *(i)* homography matrices $\mathbf{H}$ for superpixels, *(ii)* occlusion variables b, o, and *(iii)* semantic label probability maps $\mathbf{l}^{t+1}$. Optimizing the homography matrices $\mathbf{H}$ is especially challenging because the matrices have 8-DoF in 2D space and their parameterization incurs a high-dimensional search space. We address this using PMBP (Besse et al., 2013); see below for details.

Once the motion $\mathbf{H}$ is updated, occlusion variables can be easily updated independently for each pair of neighboring superpixels, while other variables are held fixed. Given their motions, we first calculate the overlapping region, which can potentially be the occluded region for one of the two superpixels. Then, we calculate the energy in Eq. (3.1) for all four boundary occlusion cases with the candidate occlusion pixels given. The boundary occlusion case that has the minimum energy is taken, including the corresponding occlusion mask state. Finally, the semantic label probability map $\mathbf{l}^{t+1}$ can also be easily updated independently for all superpixels by minimizing label data term in Eq. (3.3a).

Optimizing homography matrices using PMBP. Our method optimizes the homography matrices in the continuous domain using PMBP (Hornáček et al., 2014). PMBP is a simple but powerful optimizer based on Belief Propagation. Instead of using a discrete label set, PMBP uses a set of particles that is randomly sampled and propagated in the continuous domain. PMBP requires an effective way of proposing the random particles; typically they are obtained from a normal distribution defined over some parameters.

Algorithm 1: Optimization

initialization();
for $m = 1$ *to n-outer-iters* **do**
  for $n = 1$ *to n-inner-iters* **do**
    Optimizing $E(\mathbf{H}, \mathbf{l}^{t+1}, o, b)$ for $\mathbf{H}$ using PMBP
  end
  $\{b, o\} = \text{argmin}_{b,o}\, E(\mathbf{H}, \mathbf{l}^{t+1}, o, b)$
  $\mathbf{l}^{t+1} = \text{argmin}_{\mathbf{l}^{t+1}}\, E(\mathbf{H}, \mathbf{l}^{t+1}, o, b)$
end

In our approach, however, we devise several strategies for proposing particles of the homography matrices without over-parameterization. Between two image patches, a superpixel and its corresponding region in the other frame, we estimate the homography matrix by using *(i)* Lucas Kanade (LK) warping, *(ii)* 3 correspondences and the fundamental matrix, *(iii)* 4 randomly perturbed correspondences, and *(iv)* sampled correspondences from neighboring superpixels. Empirically, we find that these strategies generate reasonable particles without requiring an over-parameterization, and only 5 outer-iterations are enough to be converged.

3.3 EXPERIMENTS

We verify the effectiveness of our approach with a series of experiments on the well-established KITTI benchmark (Geiger et al., 2012).

We first evaluate our optical flow results on the KITTI Optical Flow 2015 benchmark (Menze et al., 2015b; 2018) and compare to the top-performing algorithms in the benchmark. In addition, we analyze the effectiveness of the semantics-related terms to understand how effectively the semantic information contributes to the estimation of optical flow. Finally, we demonstrate qualitative and quantitative results for temporally consistent semantic segmentation. We use DiscreteFlow (Menze et al., 2015a) to initialize the flow estimation and utilize the FCN model (Long et al., 2015) trained on the Cityscapes dataset (Cordts et al., 2016) for bottom-up semantic segmentation evidence. We set our parameters automatically using Bayesian optimization (Martinez-Cantin, 2014) on the training portion.

3.3.1 *KITTI 2015 optical flow*

We compare to the top-scoring optical flow methods on the KITTI Optical Flow 2015 benchmark (Menze et al., 2015b; 2018), which have been published at the time of submission. Note that we do not consider scene flow methods here, as they have access to multiple views. Table 3.1 shows the results. *Fl-bg*, *Fl-fg*, and *Fl-all* denote the flow error evaluated for background pixels only, foreground pixels only, or for all pixels, respectively. Our method outperforms all top-scoring methods when considering all non-occluded pixels and performs very close to the leading method when considering

	Non-occluded pixels			All pixels		
Method	Fl-bg	Fl-fg	Fl-all	Fl-bg	Fl-fg	Fl-all
MotionSLIC (Yamaguchi et al., 2013)	**6.19%**	64.82%	16.83%	14.86%	66.21%	23.40%
PatchBatch (Gadot and Wolf, 2016)	10.06%	26.21%	12.99%	19.98%	30.24%	21.69%
DiscreteFlow (Menze et al., 2015a)	9.96%	22.17%	12.18%	21.53%	26.68%	22.38%
SOF (Sevilla-Lara et al., 2016)	8.11%	23.28%	10.86%	**14.63%**	27.73%	**16.81%**
Ours (JFS)	7.85%	**18.66%**	**9.81%**	15.90%	**22.92%**	17.07%

Table 3.1: KITTI **Optical Flow 2015 (Menze et al., 2015b; 2018):** Comparison to the published top-performing optical flow methods in the benchmark: MotionSLIC, PatchBatch, DiscreteFlow, and SOF. Our method leads to state-of-the-art results and significantly increases the performance on challenging dynamic regions (*fg*).

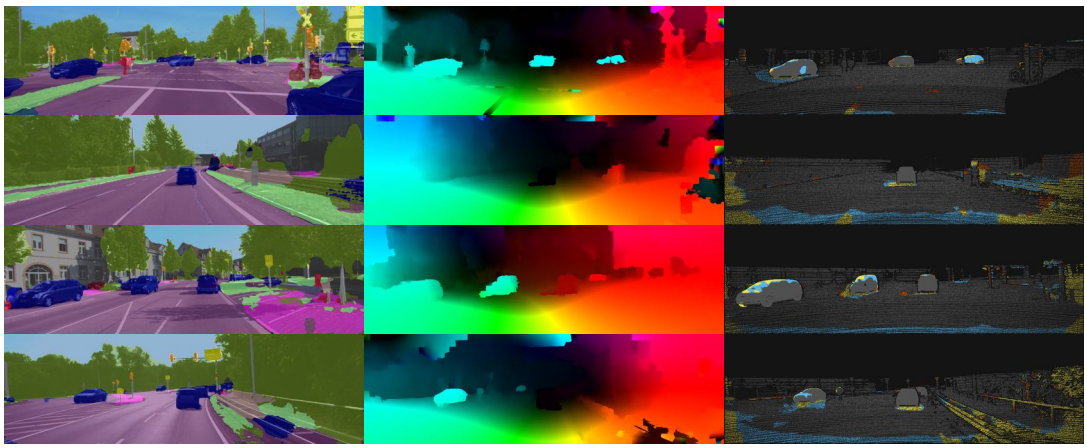


Figure 3.5: **Results on KITTI Optical Flow 2015 (Menze et al., 2015b; 2018). Left**: Source images overlaid with semantic segmentation results. **Middle**: Our flow estimation results. **Right**: Qualitative comparison with DiscreteFlow: gray pixels – both methods correct, skyblue pixels – our method is correct but DiscreteFlow is not, red pixels – DiscreteFlow is correct but ours is not, and yellow pixels – both failed.

all pixels. Especially for the flow of dynamic foreground objects, our method outperforms all published results by a large margin. This is of particular importance in the domain of autonomous navigation where understanding the motion of other traffic participants is crucial. This substantial performance gain stems from several design decisions. First, our piecewise motion representation effectively abstracts the planar surfaces of foreground vehicles, and the 8-DoF homography successfully describes the rigid motion of each surface.

The soft epipolar constraint of our model, derived from the jointly estimated semantics, contributes to the flow estimation particularly on background pixels and clear performance gains are observed for non-occluded pixels. When including occluded pixels, however, SOF (Sevilla-Lara et al., 2016) slightly outperforms ours. The main reason is that their localized layer approach and planar approximation with large pieces can regularize the occluded regions better than our piecewise

Usage of terms		Non-occluded pixels			All pixels		
Label	Epi	Fl-bg	Fl-fg	Fl-all	Fl-bg	Fl-fg	Fl-all
✓	✓	8.27%	17.40%	**9.83%**	16.44%	20.02%	**16.98%**
✓		8.45%	**16.97%**	9.90%	16.73%	**19.61 %**	17.17%
	✓	**8.20%**	17.82%	9.84%	**16.35%**	20.41%	16.99%
		8.51%	17.21%	10.00%	16.84%	19.86%	17.31%

Table 3.2: **Effectiveness of semantic-related terms:** The performance of our basic piecewise optical flow model is boosted further (KITTI 2015 training set).

model based on superpixels. In future work, this gap may be addressed through an additional global support model or coarse-to-fine estimation. MotionSLIC (Yamaguchi et al., 2013) still performs better than ours on background pixels by strictly enforcing the epipolar constraint. As a trade-off, however, their strict epipolar constraint yields significant flow errors for foreground pixels and eventually increases the overall error.

Figure 3.5 shows visual results on the KITTI dataset (visualized as in Sevilla-Lara et al. (2016)) and provides a direct comparison to DiscreteFlow, which highlights where the performance gain over the initialization originates. Our method provides more accurate flow estimates on foreground objects, but also on static objects.

3.3.2 *Effectiveness of semantic-related terms*

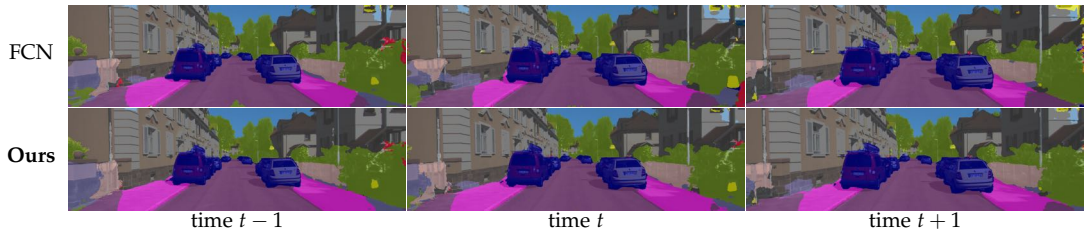
Next we analyze the effectiveness of the semantic-related terms, the epipolar constraint term and the label data term, in order to understand how much the semantic information contributes to optical flow estimation over our basic piecewise optical flow model. We turned off each term and evaluated how each setting affects the flow estimation results on the KITTI Flow 2015 training dataset. The analysis is shown in Table 3.2.

We find that the label term clearly contributes to more accurate flow estimation overall, but it has a side-effect on background areas where the initial semantic segmentation may have some outliers. Using the epipolar constraint term results in more accurate flow estimates on background areas, which majorly satisfy the epipolar assumption. On foreground objects, however, the flow error slightly increases. This performance loss is coming from the trade-off of our assumption that non-static objects (e.g., vehicles) sometimes do not move, which made us apply the epipolar cost but with a small truncation threshold.

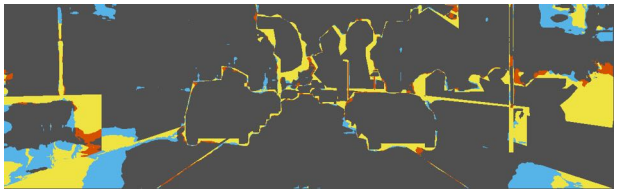
One interesting observation is that our basic piecewise flow model, without the semantic-related terms, still demonstrates competitive performance for estimating optical flow on non-occluded pixels.

IoU (%)	sky	building	road	sidewalk	fence	vegetation	pole	car	sign	pedestrian	cyclist	**mean**
FCN	69.35	78.53	73.75	38.19	33.33	68.37	**23.68**	77.60	31.27	20.11	21.42	48.69
Ours	**71.80**	**79.97**	**77.99**	**41.01**	**36.27**	**69.21**	16.44	**78.58**	**39.05**	**23.50**	**25.44**	**50.84**

Table 3.3: **Performance of temporally consistent semantic segmentation**: Our temporally consistent estimation improves the accuracy over the bottom-up segmentation results, FCN (Long et al., 2015).



(a) Results on three consecutive frames.



(b) Performance gain/loss over bottom-up semantic segmentation.

Figure 3.6: **Qualtiative example of our temporally consistent semantic segmentation** improved over the bottom-up segmentation results, FCN (Long et al., 2015).

3.3.3 *Temporally consistent semantic segmentation*

We finally evaluate the performance of our temporally consistent semantic segmentation on a sequence from the KITTI dataset, which has a 3rd-party ground truth semantic annotation (Ros et al., 2015). This, however, is a preliminary result, since the semantic segmentation model we used here is trained on the higher-resolution Cityscapes dataset (Cordts et al., 2016), which possesses somewhat different statistics. Better results are expected from a custom-trained model. Table 3.3 shows that our joint approach increases the segmentation accuracy over the bottom-up segmentation results (Long et al., 2015) by 2 percentage points in the Intersection over Union (IoU) metric. The accuracy is increased on all object classes except for the pole class, which is not well captured by our superpixels. Fig. 3.6a shows our results on three consecutive frames, and Fig. 3.6b demonstrates our performance gain/loss over the bottom-up segmentation using the visualization of Fig. 3.5. With the aid of accurate temporal correspondences, our method revises inconsistent results and effectively reduces false positives in the time domain.

3.4 DISCUSSION

We have proposed a method for the joint estimation of optical flow and temporally consistent semantic segmentation from a sequence of monouclar images. Our results on the challenging KITTI benchmark demonstrated that each task can successfully benefit from the other through the joint formulation. Based on a piecewise optical flow model with PMBP inference, which already shows highly competent results, embedding semantic information through label consistency and epipolar constraints further boosts the performance. For dynamic objects, which are particularly important in autonomous navigation research, our method substantially outperforms all prior studies reported in the benchmark. Preliminary results on temporally consistent semantic segmentation further demonstrate the benefit of our approach by reducing false positives and flickering. However, our method relies on somewhat strong assumptions that can lead potential limitations of our method. Unless reliable semantic bottom-up evidence is given, noisy semantic cue may cause inaccuracy on flow estimation due to the strong epipolar constraints for static objects and the semantic consistency term. Furthermore, inaccuracy on the superpixelization (*e. g.* when a superpixel lies on two diffent moving objects.) would also cause errors on the flow estimation because our method estimates one 8-DoF homography motion per each superpixel. As future work, we believe that learning-based approaches can resolve such limitations, *e. g.*, by learning to compensate noises on the input signal. For joint optical flow and occlusion estimation, we demonstrate a learning-based approach in Chapter 5, which advances a energy-based approach in Chapter 4.

4

A SYMMETRIC APPROACH TO JOINT OPTICAL FLOW AND OCCLUSION ESTIMATION

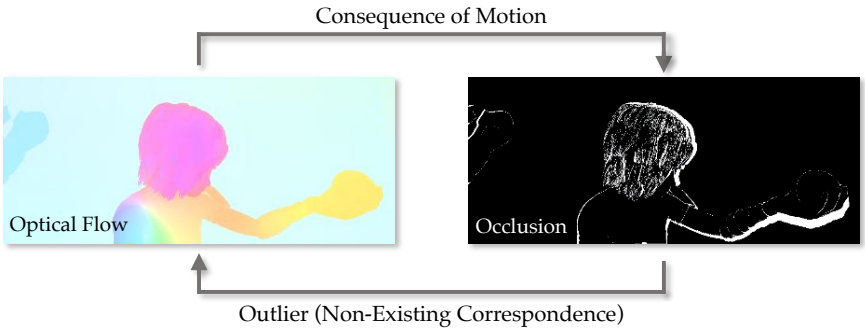


Figure 4.1: **Chicken-and-egg relationship between optical flow and occlusion**: Accurate knowledge of occluded areas is crucial for reliable optical flow estimation. Yet conversely, accurate estimation of optical flow is required for localizing occlusions reliably.

CONTENTS

Optical flow estimation is one of the most studied topics in computer vision; yet, recent benchmark datasets continue to question today's approaches, leaving occlusions as one of the key challenges. As our second joint objective with motion estimation, in this chapter, we propose a joint estimation approach to optical flow

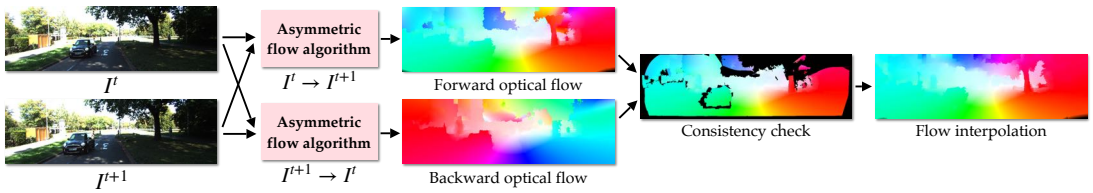
and occlusion by addressing the well-known chicken-and-egg relation between the two objectives. In contrast to many state-of-the-art methods that consider occlusions as outliers, possibly filtered out during post-processing, we highlight the importance of joint occlusion reasoning in the optimization and show how to utilize occlusion as an important cue for optical flow estimation. The key feature of our model is to fully exploit the symmetry properties that characterize optical flow and occlusions in the two consecutive images. Specifically through utilizing forward-backward consistency and occlusion-disocclusion symmetry in the energy, our model jointly estimates optical flow in both forward and backward directions, as well as consistent occlusion maps in both views. We demonstrate significant performance benefits on standard benchmarks, especially from the occlusion-disocclusion symmetry. On the challenging KITTI benchmark, we report the most accurate two-frame results at the time of publication.

4.1 INTRODUCTION

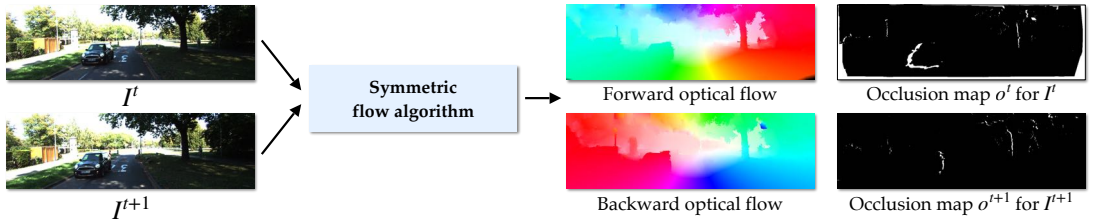
Optical flow estimation has been studied extensively for several decades. Yet, recent optical flow benchmark datasets (Butler et al., 2012; Menze et al., 2015b; 2018) reveal challenges that current methods still have. Occlusion is one of them alongside complex motion and severe illumination changes. Thus, proper occlusion handling, especially in the presence of large motion, plays a critical role in determining the extent of a method's success in challenging scenes.

Occlusion estimation is a well-known chicken-and-egg problem that optical flow has confronted for a long time (Alvarez et al., 2007; Kolmogorov and Zabih, 2001; Pérez-Rúa et al., 2016; Strecha et al., 2004). As illustrated in Fig. 4.1, accurate knowledge of occluded areas is crucial for reliable optical flow estimation in order to prevent non-occluded areas from adverse effects by occluded pixels. Occlusion is a consequence of motion; in turn, estimation of accurate optical flow, conversely, is required for reliable localization of occlusions. We thus posit that their mutual dependence necessitates taking a joint approach for which no prior studies have explored to the extent possible.

As discussed in Section 2.3.2, the majority of recent work instead addresses this challenging problem indirectly, considering occlusions *outliers* of low-level correspondence estimation (Bailer et al., 2015; Chen and Koltun, 2016; Güney and Geiger, 2016; Hu et al., 2016; Yang and Li, 2015). Such approaches aim to mitigate the effects of occlusion by exploiting that occluded pixels generally violate the underlying model assumptions as there are no corresponding pixels in the other frame. The use of a robust, truncated penalty in the data term naturally reduces the effects of the high data cost from occlusions, but also more generally from outlier pixels that violate brightness constancy (Chen and Koltun, 2016; Menze et al., 2015a; Yang and Li, 2015). As shown in Fig. 4.2a, those methods then separately estimate forward and backward flow, check the forward-backward motion consistency in a subsequent post-processing step, and extrapolate flow into inconsistent regions, which help resolve the motion mismatch in the occluded area (Chen and Koltun, 2016; Gadot and Wolf, 2016; Hu



(a) A conventional asymmetric approach



(b) Our integrative, symmetric approach

Figure 4.2: Comparison between *(a)* a conventional asymmetric approach that requires post-processing and *(b)* Our integrative, symmetric approach.

et al., 2016; Menze et al., 2015a) in the end. These simple procedures have been shown to diminish the influence of occlusions in practice.

Such strategies, however, still cannot completely rule out the adverse effects of occlusion in optical flow estimation. Using a truncated data term leaves the possibility that occluded pixels can be incorrectly matched to other pixels for which the data cost is lower than the truncation constant. Additionally, when extrapolating flow in post-processing, false-positive matches may remain even after the forward-backward consistency check and can cause erroneous estimates to be propagated across a local region (Revaud et al., 2015). We thus argue that only the accurate localization of occluded regions, formulated as a joint estimation together with the flow, can fundamentally resolve this intertwined problem.

In this chapter, we address the chicken-and-egg problem of optical flow and occlusion map estimation and propose a joint energy formulation and optimization method. As shown in Fig. 4.2b, our approach directly utilizes their relationship and allows them to leverage each other through forward and backward flow estimation as well as occlusion maps for both directions altogether, as shown in Fig. 4.3. We exploit two key symmetry properties of the optical flow field and the occlusion map within the two consecutive images: forward-backward flow consistency and occlusion-disocclusion symmetry. These symmetry properties not only couple optical flow with occlusion, but also allow to exploit the geometric and temporal information in the two consecutive images to a greater extent. Optical flow in occluded area cannot be accuractely estimated due to non-existence of its correspondence, but our method infers more accurate flow for occluded area than previous methods through better occlusion localization and piece-wise smoothness.

The key contributions of this chapter are as follows. To the best of our knowledge, we are the first to exploit the occlusion-disocclusion symmetry in joint optical flow and occlusion estimation and show its significant accuracy gains. Second, we

Figure 4.3: **Results of our symmetric optical flow approach given two consecutive images from the KITTI benchmark** (Menze et al., 2015b; 2018). Our method jointly predicts forward and backward optical flow *(overlaid on top row, from left to right)* as well as corresponding occlusion maps for each view *(overlaid on bottom row)*.

demonstrate how this joint, symmetric treatment combined with a piecewise rigid formulation allows optical flow estimation without post-processing. At the time of publication, our experimental results demonstrated state-of-the-art accuracy on public benchmark datasets, where we improve the results especially in occluded areas. For the challenging KITTI dataset, we reported the most accurate results among two-frame methods, outperforming the approaches based on high-capacity deep networks (Bai et al., 2016; Dosovitskiy et al., 2015; Güney and Geiger, 2016; Ilg et al., 2017). The fact that we are able to do so without employing learning demonstrates the significant benefits and strengths of our joint, symmetric flow and occlusion formulation.

4.2 JOINT, SYMMETRIC APPROACH

From two consecutive images, we jointly estimate optical flow maps in both directions and corresponding occlusion maps by fully exploiting their symmetries. The symmetry properties *(i)* couple the two different problem domains, *(ii)* better utilize the available image evidence, and *(iii)* lead to a well-balanced solution during optimization.

The first symmetry we consider is bi-directional motion consistency, *i. e.* motions of corresponding pixels that are visible in both views should be the inverse of one another. Unlike most previous work, we integrate this consistency in the energy, which not only leads to better estimates of both forward and backward flow through iterative optimization, but also obviates conventional post-processing.

Occlusion-disocclusion symmetry is the second property we consider, which geometrically explains how occlusions arise from differing motions of dynamic entities. As illustrated in the third and fourth column of Fig. 4.4, occlusions and disocclusions demonstrate a symmetry relationship, *i. e.* occlusions in the forward direction correspond to disocclusions in the backward direction and vice versa.

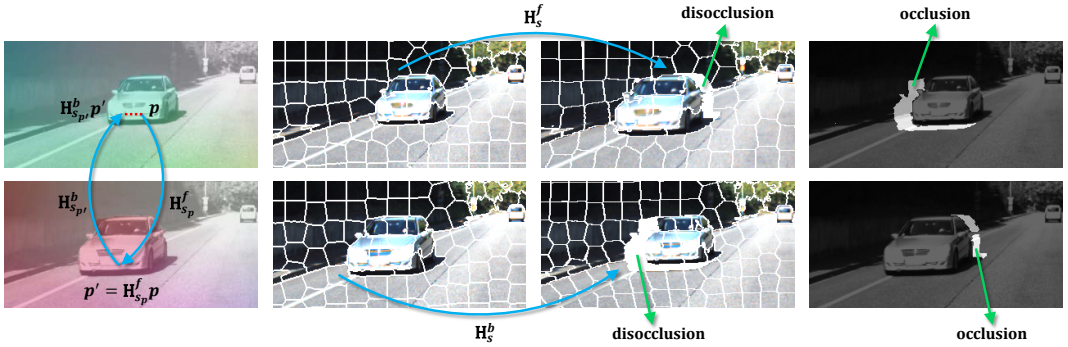


Figure 4.4: Conceptual explanation of our approach: *(top to bottom)* first frame I^t, second frame I^{t+1}; *(left to right)* forward/backward flow maps overlaid on each source image, raw superpixel images, warping results given homography motions $\mathbf{H}$, and occlusion maps.

4.2.1 *Piecewise rigid optical flow model*

Our optical flow model is based on a piecewise rigid representation, which has recently been found to allow the effective regularization of 8-DoF or 9-DoF rigid motion of entities in images, yet still represent both diverse and general motions (Hornáček et al., 2014; Hur and Roth, 2016; Menze and Geiger, 2015; Vogel et al., 2013b; Yang and Li, 2015). We first decompose an image into a set of superpixels (Yao et al., 2015) as shown in Fig. 4.4, and estimate the 8-DoF homography motion $\mathbf{H}$ of each superpixel. Each superpixel represents a possible surface in the scene, and the homography represents a locally rigid motion of the surface. Using this model also facilitates formulating the occlusion-disocclusion symmetry property in a comprehensive way, which will be explained below.

4.2.2 *Joint energy with symmetries*

Given the two consecutive images I^t and I^{t+1} with their superpixel representations, our model jointly estimates *(i)* the forward motion $\mathbf{H}^f$ (*i.e.*, $I^t \rightarrow I^{t+1}$) and the backward motion $\mathbf{H}^b$ (*i.e.*, $I^{t+1} \rightarrow I^t$) of each superpixel, and *(ii)* the per-pixel occlusion maps o^t and o^{t+1} for each view. We formulate this through the energy

$$\begin{aligned} E(\mathbf{H}^f, \mathbf{H}^b, o^t, o^{t+1}) = {} & E_{\mathrm{D}}(\mathbf{H}^f, \mathbf{H}^b, o^t, o^{t+1}) \\ & + \lambda_{\mathrm{P}} E_{\mathrm{P}}(\mathbf{H}^f, \mathbf{H}^b, o^t, o^{t+1}) \\ & + \lambda_{\mathrm{C}} E_{\mathrm{C}}(\mathbf{H}^f, \mathbf{H}^b, o^t, o^{t+1}) \\ & + \lambda_{\mathrm{S}} E_{\mathrm{S}}(o^t, o^{t+1}), \end{aligned} \tag{4.1}$$

which consists of a data term E_{D}, a pairwise term E_{P}, a forward-backward consistency term E_{C}, and an occlusion-disocclusion symmetry term E_{S}.

Data term. The data term accumulates photometric differences across all pixels in both views given the 8-DoF homography motions of superpixels $\mathbf{H}$ and the per-pixel occlusion masks $o_{\mathbf{p}}$ in both views:

$$E_{\mathrm{D}}(\mathbf{H}^f, \mathbf{H}^b, o^t, o^{t+1}) = \sum_{\mathbf{p} \in I^t} D^f_{\mathbf{p}} + \sum_{\mathbf{p} \in I^{t+1}} D^b_{\mathbf{p}} \tag{4.2a}$$

with

$$D^f_{\mathbf{p}} = \overline{o^t_{\mathbf{p}}}\, \rho^f_{\mathrm{D}}(\mathbf{p}, \mathbf{H}^f_{s_{\mathbf{p}}}) + o^t_{\mathbf{p}} \lambda_{\mathrm{occ}} \tag{4.2b}$$

$$D^b_{\mathbf{p}} = \overline{o^{t+1}_{\mathbf{p}}} \rho^b_{\mathrm{D}}(\mathbf{p}, \mathbf{H}^b_{s_{\mathbf{p}}}) + o^{t+1}_{\mathbf{p}} \lambda_{\mathrm{occ}}. \tag{4.2c}$$

For a non-occluded pixel $\mathbf{p}$ (*i. e.*, $\overline{o_{\mathbf{p}}} = (1 - o_{\mathbf{p}}) = 1$), the function $\rho_{\mathrm{D}}(\mathbf{p}, \mathbf{H}_{s_{\mathbf{p}}})$ measures the truncated photometric error between pixel $\mathbf{p}$ and its corresponding pixel $\mathbf{H}_{s_{\mathbf{p}}}\mathbf{p}$ in the other frame. $\mathbf{H}_{s_{\mathbf{p}}}$ is the homography motion of superpixel $s_{\mathbf{p}}$ at pixel position $\mathbf{p}$. We use a weighted sum of a gradient constancy term and a ternary transform (Stein, 2004), which is known to be robust under illumination changes (Hafner et al., 2013; Vogel et al., 2013a; b). When the corresponding location $\mathbf{H}_{s_{\mathbf{p}}}\mathbf{p}$ falls outside the image boundary, $\rho_{\mathrm{D}}(\cdot, \cdot)$ outputs the truncation constant τ_{D}.

More specifically, we use a continuous version of the ternary transform, which can improve localization (Vogel et al., 2013a) compared to the conventional discrete setting. Furthermore, when calculating the ternary value in the other frame, we transform 7×7 patches from the reference frame to the other using the given homography $\mathbf{H}_{s_{\mathbf{p}}}$, and calculate the ternary transform based on the transformed patches. Using this strategy yields a more comprehensive data cost that is invariant to local shape deformation caused by the motion. We observe this to increase the flow accuracy; see Appendix A.1 for details and quantitative analysis.

For occluded pixels (*i. e.*, $o_{\mathbf{p}} = 1$), the constant penalty λ_{occ} is applied so that we can avoid trivial cases in which all pixels are occluded or move outside the image boundary. We set $\lambda_{\mathrm{occ}} < \tau_{\mathrm{D}}$ so that pixels whose corresponding location is outside of the image boundary can be naturally inferred as occluded pixels during the optimization. Although occlusions are treated as outliers (*i. e.* the constant penalty) in the data term, they serve as important visual cues and indirectly help estimate optical flow through the occlusion-disocclusion symmetry term, which we present below.

Pairwise term. The pairwise term penalizes the motion differences and occlusion status differences in an 8-neighborhood $N(\mathbf{p})$:

$$E_{\mathrm{P}}(\mathbf{H}^f, \mathbf{H}^b, o^t, o^{t+1}) = \sum_{\mathbf{p} \in I^t} P^f_{\mathbf{p}} + \sum_{\mathbf{p} \in I^{t+1}} P^b_{\mathbf{p}} \tag{4.3a}$$

with

$$P^f_{\mathbf{p}} = \sum_{\mathbf{q} \in N(\mathbf{p})} \left(\phi(\mathbf{H}^f_{s_{\mathbf{p}}}, \mathbf{H}^f_{s_{\mathbf{q}}}, \bar{\mathbf{p}}) + \lambda_{\mathrm{O}} \left[o^t_{\mathbf{p}} \neq o^t_{\mathbf{q}} \right] \right) \tag{4.3b}$$

$$P^b_{\mathbf{p}} = \sum_{\mathbf{q} \in N(\mathbf{p})} \left(\phi(\mathbf{H}^b_{s_{\mathbf{p}}}, \mathbf{H}^b_{s_{\mathbf{q}}}, \bar{\mathbf{p}}) + \lambda_{\mathrm{O}} \left[o^{t+1}_{\mathbf{p}} \neq o^{t+1}_{\mathbf{q}} \right] \right), \tag{4.3c}$$

where λ_{O} is a weight for the pairwise occlusion cost and $[\cdot]$ denotes the Iverson bracket. $\phi(\mathbf{H}_{s_{\mathbf{p}}}, \mathbf{H}_{s_{\mathbf{q}}}, \bar{\mathbf{p}})$ measures the difference of two motions induced by the homographies

$\mathbf{H}_{s_\mathbf{p}}$ and $\mathbf{H}_{s_\mathbf{q}}$ at neighboring pixels $\mathbf{p}$ and $\mathbf{q}$, evaluated in the middle between the pixels $\bar{\mathbf{p}} = \frac{\mathbf{p}+\mathbf{q}}{2}$. This function distinguishes three different types of pairwise relationships (Hur and Roth, 2016; Yamaguchi et al., 2013; 2014) in order to effectively express geometric relations that two neighboring superpixels can have (*e. g.*, co-planar, hinge, others) in terms of their homography motion:

$$\phi(\mathbf{H}_{s_\mathbf{p}}, \mathbf{H}_{s_\mathbf{q}}, \bar{\mathbf{p}}) = w_{\mathbf{p},\mathbf{q}} \cdot \min(\phi_{\text{co}}, \phi_{\text{h}}, \tau_{\text{P}}) \tag{4.4a}$$

with

$$\phi_{\text{co}} = \frac{1}{|s_\mathbf{p} \cup s_\mathbf{q}|} \sum_{\mathbf{p}_i \in s_\mathbf{p} \cup s_\mathbf{q}} \|\mathbf{H}_{s_\mathbf{p}} \mathbf{p}_i - \mathbf{H}_{s_\mathbf{q}} \mathbf{p}_i\| \tag{4.4b}$$

$$\phi_{\text{h}} = \|\mathbf{H}_{s_\mathbf{p}} \bar{\mathbf{p}} - \mathbf{H}_{s_\mathbf{q}} \bar{\mathbf{p}}\| + \lambda_{\text{h}}, \tag{4.4c}$$

where τ_{P} is a truncation constant and λ_{h} a constant bias. The intensity-adaptive weight $w_{\mathbf{p},\mathbf{q}} = \exp(-|I(\mathbf{p}) - I(\mathbf{q})|/\sigma_w)$ controls the strength of the pairwise term depending on the intensity difference between two neighboring pixels.

The co-planar potential ϕ_{co} calculates the average difference of the homography motions for all pixels within the union of the two superpixels, as they are on the same plane by assumption. The hinge potential ϕ_{h} penalizes the motion difference only for the middle pixel and applies a constant bias λ_{h}. For handling the remaining cases such as occlusion or disocclusion, the truncation constant τ_{P} is used. Note that the pairwise cost only becomes effective between neighboring pixels that belong to two different superpixels. The cost between neighboring pixels in the same superpixel is naturally zero, because the two pixels have the same associated homography motion.

The pairwise occlusion term simply encourages the spatial smoothness of the occlusion states by penalizing their differences between neighboring pixels. If neighboring pixels have differing occlusion states, the term incurs the constant penalty λ_{O}.

Forward-backward consistency term. Unlike conventional optical flow algorithms, our model explicitly integrates forward-backward consistency[1] into the objective function (Ince and Konrad, 2008):

$$E_{\text{C}}(\mathbf{H}^f, \mathbf{H}^b, o^t, o^{t+1}) = \sum_{\mathbf{p} \in I^t} C^f_\mathbf{p} + \sum_{\mathbf{p} \in I^{t+1}} C^b_\mathbf{p} \tag{4.5a}$$

with

$$C^f_\mathbf{p} = \overline{o^t_\mathbf{p}}\ \overline{o^{t+1}_{\mathbf{p}'}} \rho_{\text{C}}(\|\mathbf{p} - \mathbf{H}^b_{s_{\mathbf{p}'}} \mathbf{H}^f_{s_\mathbf{p}} \mathbf{p}\|) \tag{4.5b}$$

$$C^b_\mathbf{p} = \overline{o^{t+1}_\mathbf{p}}\ \overline{o^t_{\mathbf{p}''}} \rho_{\text{C}}(\|\mathbf{p} - \mathbf{H}^f_{s_{\mathbf{p}''}} \mathbf{H}^b_{s_\mathbf{p}} \mathbf{p}\|), \tag{4.5c}$$

where $\mathbf{p}' = \mathbf{H}^f_{s_\mathbf{p}} \mathbf{p}$ and $\mathbf{p}'' = \mathbf{H}^b_{s_\mathbf{p}} \mathbf{p}$. The consistency term penalizes the Euclidean distance between the position of pixel $\mathbf{p}$ and the back-projected position of the corresponding pixel in the other frame, as illustrated in the first column of Fig. 4.4. Thus, the term encourages the homography matrices from the two corresponding

1 This forward-backward consistency, by the way, also appears as a cycle consistency in other literatures, *e. g.*, generative models (Zhu et al., 2017a), domain adaptation (Hoffman et al., 2018), or self-supervised representation learning (Wang et al., 2019a; Xiong et al., 2021), by imposing a cyclic consistency from bi-directional outputs and using it as a self-supervised learning signal.

points to be under an inverse relationship, providing a soft constraint on motions in the opposite view, which improves the estimation as shown below. The function $\rho_C(\cdot)$ truncates its input at τ_C to be robust to possible outliers of flow or occlusion.

The term takes into account all pixels in both views, but applies the penalty only if the pixel $\mathbf{p}$ and its corresponding pixel $\mathbf{p}'$ or $\mathbf{p}''$ in the respective other frame are not occluded. This condition enforces pixels to be either occluded or their motion to satisfy the bi-directional symmetry property. The condition may lead to a trivial solution in which all pixels are marked as becoming occluded such that no penalties arise from this term. However, the constant penalty λ_{occ} in the data term, *cf*. Eq. (4.2b) and Eq. (4.2c), prevents the solution from falling into this trivial case and balances between visible pixels and occlusions.

Occlusion-disocclusion symmetry term. The occlusion-disocclusion symmetry term plays the most important role in our model, and allows flow and occlusions to mutually leverage one another. Specifically, the term penalizes cases in which the occlusion-disocclusion symmetry relationship does not hold in both views:

$$E_S(o^t, o^{t+1}) = \sum_{\mathbf{p} \in I^t} o_{\mathbf{p}}^t \odot N_{\mathbf{p}}^t + \sum_{\mathbf{p} \in I^{t+1}} o_{\mathbf{p}}^{t+1} \odot N_{\mathbf{p}}^{t+1} \tag{4.6a}$$

with

$$N_{\mathbf{p}}^t = \left| \left\{ \mathbf{p} \mid \mathbf{p} = \mathbf{H}_{s_{\mathbf{p}'}}^b \mathbf{p}',\ \forall \mathbf{p}' \in I^{t+1} \right\} \right| \tag{4.6b}$$

$$N_{\mathbf{p}}^{t+1} = \left| \left\{ \mathbf{p} \mid \mathbf{p} = \mathbf{H}_{s_{\mathbf{p}'}}^f \mathbf{p}',\ \forall \mathbf{p}' \in I^{t} \right\} \right|. \tag{4.6c}$$

The XNOR operation (*e.g.*, $0 \odot 0 = 1$ and $1 \odot 0 = 0$) ensures that if a pixel $\mathbf{p}$ is being occluded in one frame, then there cannot be any pixels in the other frame whose motion maps to $\mathbf{p}$. We explicitly detect disocclusion in each view, representing it with the variables $N_{\mathbf{p}}^t$ and $N_{\mathbf{p}}^{t+1}$, respectively. $N_{\mathbf{p}}^t$ denotes the number of pixels that are mapped to $\mathbf{p}$ in I^t when warping pixels in I^{t+1} to I^t given their corresponding motion $\mathbf{H}_s^b$. $N_{\mathbf{p}}^{t+1}$ is defined analogously by warping pixels from I^t to I^{t+1} given the set of per-superpixel homography motions $\mathbf{H}_s^f$. If no pixel is mapped to $\mathbf{p}$ in I^{t+1} (*i.e.*, $N_{\mathbf{p}}^{t+1} = 0$), the pixel $\mathbf{p}$ is being disoccluded, as visualized in the third column of Fig. 4.4. By defining disocclusions solely as a result of the motion, their corresponding variables do not need to be optimized directly, but are indirectly determined by the motion $\mathbf{H}_s^b$ and $\mathbf{H}_s^f$. During the optimization, for example, occlusion at the current frame indirectly helps estimate backward optical flow by encouraging the correspondence between the occlusion and disocclusion detected from the backward optical flow. Vice versa, the detected disocclusion from the backward optical flow also helps better localization of the occlusion at the current frame as well.

In previous studies (Kolmogorov and Zabih, 2001; Unger et al., 2012), similar concepts have been introduced in different forms by encouraging unique correspondences between pixels visible in both views. Instead, we model symmetry between occlusions and disocclusions while allowing multiple pixels to be mapped to a single location (*i.e.*, $N_{\mathbf{p}}$ may be greater than 1), which is important when objects in the scene change their apparent size. Moreover, we perform spatial regularization of occlusion states, *cf*. Eqs. (4.3b) and (4.3c).

Algorithm 2: Optimization

initialize optical flow and occlusion maps
for $n = 1$ *to max-iteration* **do**
  // estimate forward flow
  $\mathbf{H}^f = \text{argmin}_{\tilde{\mathbf{H}}^f} E(\tilde{\mathbf{H}}^f, \mathbf{H}^b, o^t, o^{t+1})$
  // estimate occlusion map at time $t+1$
  update N^{t+1}
  $o^{t+1} = \text{argmin}_{\tilde{o}^{t+1}} E(\mathbf{H}^f, \mathbf{H}^b, o^t, \tilde{o}^{t+1})$
  // estimate backward flow
  $\mathbf{H}^b = \text{argmin}_{\tilde{\mathbf{H}}^b} E(\mathbf{H}^f, \tilde{\mathbf{H}}^b, o^t, o^{t+1})$
  // estimate occlusion map at time t
  update N^t
  $o^t = \text{argmin}_{\tilde{o}^t} E(\mathbf{H}^f, \mathbf{H}^b, \tilde{o}^t, o^{t+1})$
end

Algorithm 3: Optimizing $\mathbf{H}^f$

INPUT: a current solution of $\mathbf{H}^f$
// local expansion move
for *each local region* R_i **do**
  $\{\mathbf{H}_s^f\} \leftarrow \text{propagation}(\{\mathbf{H}_s^f \mid s \in R_i\})$
  $\{\mathbf{H}_s^f\} \leftarrow \text{randomization}(\{\mathbf{H}_s^f \mid s \in R_i\})$
end
// global expansion move
$\mathbf{H}^f \leftarrow \text{propagation}(\mathbf{H}^f)$

4.2.3 *Optimization*

We jointly optimize the two different sets of variables – homography motions (*i.e.*, $\mathbf{H}^f$ and $\mathbf{H}^b$) and occlusion maps (*i.e.*, o^t and o^{t+1}) – using a block coordinate descent algorithm, which optimizes the variables alternatingly. As described in Alg. 2, we first estimate the forward flow $\mathbf{H}^f$. Then, we update N^{t+1} from forward flow $\mathbf{H}^f$ and estimate the occlusion map o^{t+1}. The remaining variables $\mathbf{H}^b$, N^t, and o^t are updated in turn in a similar manner.

Optimizing $\mathbf{H}^f$ and $\mathbf{H}^b$. Due to the difficulty of optimizing the continuous variables (*e.g.*, stemming from the nonlinearity of the data term), we instead solve a discrete multi-label optimization problem that collects a number of candidate homography motions as proposal sets and then chooses the most suitable motion for each superpixel using fusion moves or α-expansion with Quadratic Pseudo-Boolean Optimization (QPBO) (Lempitsky et al., 2008; Rother et al., 2007; Taniai et al., 2014; Vogel et al., 2013b). For an efficient optimization, we sequentially run expansion moves locally on subgraphs of superpixels (*e.g.*, a set of neighboring 30 superpixels with 70%

overlap between each other)[2] and then globally on all the superpixels as described in Alg. 3. This strategy follows the similar approach in Taniai et al. (2017), which defines subgraphs on multiple scales and optimizes them sequentially. We found that this combined local and global optimization strategy yields a faster convergence while avoiding local minima, especially for the planar surface representation with homography-parameterized motions as used here.

The local expansion moves on subgraphs of superpixels (Taniai et al., 2017) consist of two steps, *propagation* and *randomization*, which spatially propagate homography motions and locally refine them. The global expansion moves only conduct the *propagation* step that spatially propagates the locally-refined motions into a broader area.

In each *propagation* and *randomization* step in Alg. 3, we run expansion moves on the set of input superpixels with each collected set of homography motions. The *propagation* and *randomization* step only differ in how they collect the proposal sets.

propagation:

- Randomly sampling n_p homography motions from the input set.
- Randomly sampling n_p homography motions from the corresponding superpixels in the opposite view and taking the inverse motion.

randomization:

- Randomly sampling n_p homography motions from the input set and adding perturbations.
- Sampling n_p homography motions by randomly sampling n_r point correspondence pairs and re-estimating the corresponding homography motion.

We set $n_p = 6$ and $n_r = 20$ for the local expansion moves and $n_p = 50$ for the global expansion moves.

Optimizing o^t and o^{t+1}. Optimizing the occlusion maps is relatively simple. For instance, once N^t is updated from the backward flow $\mathbf{H}^b$, the binary occlusion map o^t can be estimated via graph-cuts (Boykov and Kolmogorov, 2004; Boykov et al., 2001; Kolmogorov and Zabih, 2004) while holding all other variables fixed. The pairwise occlusion term in Eqs. (4.3b) and (4.3c) is submodular, thus making standard graph-cuts applicable.

4.3 EXPERIMENTS

We evaluate the accuracy of our algorithm on the KITTI Optical Flow 2015 benchmark (Menze et al., 2015b; 2018) and on the MPI Sintel Flow Dataset (Butler et al., 2012), both qualitatively and quantitatively. Additionally, we analyze the importance of our symmetric approach by turning each term off and evaluating how significantly it affects the accuracy. For faster convergence, we use DiscreteFlow (Menze et al., 2015a) to initialize our estimation as well as to derive proposals. We automatically tune the parameters (weights $\lambda_P, \lambda_C, \lambda_S, \lambda_O$, truncation thresholds τ_D, τ_P, τ_C, and biases

2 See appendix in Appendix A.2 for an analysis of this design choice.

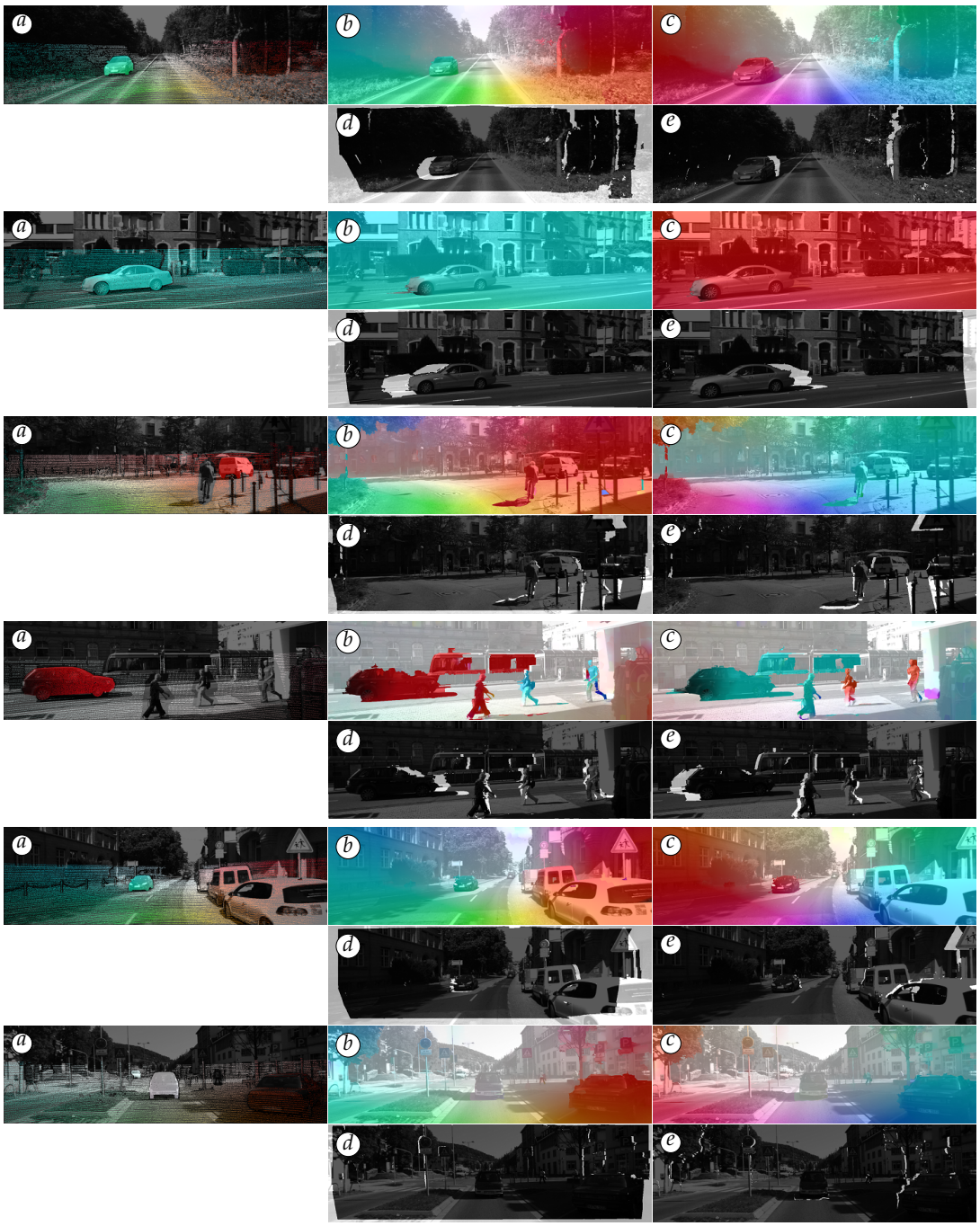


Figure 4.5: **Qualitative results on KITTI 2015**: *(a)* ground truth flow, *(b)* forward flow, *(c)* backward flow, *(d)* occlusion map on the current frame, and *(e)* occlusion map on the next frame; all overlayed with the current and next frame respectively. Note that the ground truth flow map on KITTI is sparse, and some objects are masked.

λ_{occ}, λ_h) using Bayesian optimization (Martinez-Cantin, 2014) on the training portion of each benchmark.

	Non-occluded pixels			All pixels		
Method	Fl-bg	Fl-fg	Fl-all	Fl-bg	Fl-fg	Fl-all
Ours (MirrorFlow)	6.24%	12.95%	7.46%	8.93%	17.07%	**10.29%**
FlowNet2 (Ilg et al., 2017)	7.24%	**5.60%**	**6.94%**	10.75%	**8.75%**	10.41%
SDF (Bai et al., 2016)	**5.75%**	18.38%	8.04%	**8.61%**	23.01%	11.01%
MR-Flow (Wulff et al., 2017)	6.86%	17.91%	8.86%	10.13%	22.51%	12.19%
DCFlow (Xu et al., 2017)	8.04%	19.84%	10.18%	13.10%	23.70%	14.86%
SOF (Sevilla-Lara et al., 2016)	8.11%	18.16%	9.93%	14.63%	22.83%	15.99%
DiscreteFlow (Menze et al., 2015a)	9.96%	17.03%	11.25%	21.53%	21.76%	21.57%

Table 4.1: KITTI **Optical Flow 2015:** Comparison to top-performing optical flow algorithms in the benchmark in terms of percentages of pixels with an incorrect flow estimate. It is considered incorrect if flow end-point error exceeds 3 pixels and 5%. The best and the second best results are in bold and underlined, respectively.

4.3.1 *KITTI Optical Flow 2015*

On the KITTI Optical Flow 2015 benchmark, our MirrorFlow algorithm outperforms all two-frame optical flow methods at the time of writing, demonstrating the lowest percentage of flow outliers (*Fl-all*). Table 4.1 gives detailed numbers, Fig. 4.5 shows qualitative results. Leveraging our symmetry terms and motion representation based on piecewise homographies, our method also demonstrates the second best results for handling flow on dynamic foreground objects (*Fl-fg*) and handling background motion (*Fl-bg*).

For evaluating the accuracy especially in occluded areas, we cannot directly rely on the KITTI accuracy indicators, as the number of occluded pixels and non-occluded pixels in the testing dataset are unknown. However, by looking at the gap of outlier percentages between all pixels and non-occluded pixels, we can infer that our symmetric method is among the most accurate (probably the most accurate) method in occluded areas (see Appendix A.3 for a more detailed discussion).

One important remark is that our method does not use any learned feature descriptors or semantic information unlike other top-performing algorithms (Bai et al., 2016; Ilg et al., 2017; Sevilla-Lara et al., 2016; Wulff et al., 2017; Xu et al., 2017). Even without them, our method significantly outperforms the baseline method (Menze et al., 2015a), which is the next best method not relying on learned descriptors or semantics, and is also used for initialization. We significantly reduce the number of incorrect pixels by more than 50% by virtue of our joint, symmetric formulation. We believe that our method still has room for substantial further improvement by exploiting semantic information or using learned feature descriptors; we leave this for future work.

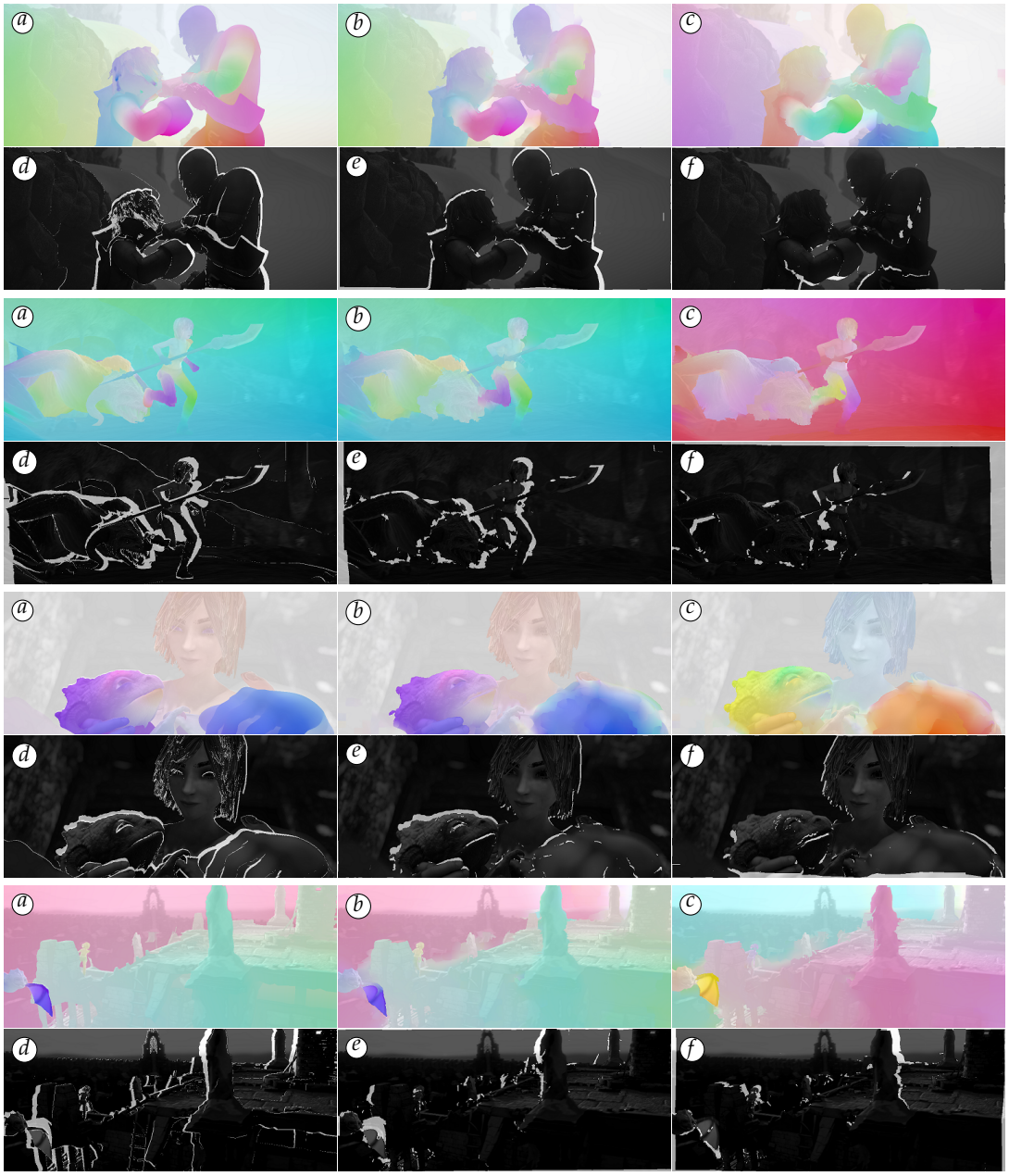


Figure 4.6: **Qualitative results on Sintel**: *(a)* ground truth flow, *(b)* forward flow, *(c)* backward flow, *(d)* ground truth occlusion, *(e)* occlusion map on the current frame, and *(f)* occlusion map on the next frame; all overlayed with the current and next frame respectively. Note the high agreement between true and estimated occlusions.

4.3.2 *MPI Sintel Flow Dataset*

While we focus on the KITTI dataset with its challenging scenes in the context of autonomous driving, we also evaluate our approach on the MPI Sintel Flow Dataset (Butler et al., 2012), where approaches based on piecewise rigidity are known to be somewhat disadvantaged. Nevertheless, our method still performs rather competi-

	Final pass			Clean pass		
Method	EPE all	EPE nocc.	EPE occ.	EPE all	EPE nocc.	EPE occ.
DCFlow (Xu et al., 2017)	**5.119**	**2.283**	28.228	3.537	1.103	23.394
FlowFieldsCNN (Bailer et al., 2017)	5.363	2.303	30.313	3.778	0.996	26.469
MR-Flow (Wulff et al., 2017)	5.376	2.818	**26.235**	**2.527**	**0.954**	**15.365**
S2F-IF (Yang and Soatto, 2017)	5.417	2.549	28.795	3.500	0.988	23.986
RicFlow (Hu et al., 2017)	5.620	2.765	28.907	3.550	1.264	22.220
GlobalPatchCollider (Wang et al., 2016)	6.040	2.938	31.309	4.134	1.432	26.179
Ours (MirrorFlow)	6.071	3.186	29.567	3.316	1.338	19.470
DiscreteFlow (Menze et al., 2015a)	6.077	2.937	31.685	3.567	1.108	23.626

Table 4.2: MPI **Sintel Flow Dataset:** Accuracy in terms of the average End-Point Error (EPE). Leading algorithms on Final or Clean. Our model performs the second best in the Clean pass.

tively, achieving the second place in the Clean pass and the 13th place in the Final pass, both at the time of writing. Table 4.2 gives detailed accuracy numbers. Fig. 4.6 shows qualitative results using the color code of the Sintel dataset, where we observe rather accurate estimates of occluded regions. The main reason why our method is not as accurate as on the KITTI benchmark is that our planar-rigid motion assumption is not as appropriate for the Sintel Dataset, where the majority of motions are non-rigid and most surfaces are non-planar. Despite of this limitation, our method demonstrates leading results especially on occluded pixels in the Clean pass, which once again confirms the key benefits of our joint flow and occlusion estimation pipeline. Note that while we do not consider this here, the key ideas behind our joint, symmetric approach are not limited to piecewise rigid representations.

4.3.3 *Importance of symmetries*

We conduct an ablation study to emphasize the contribution of our symmetric formulation and to demonstrate how much each symmetry property contributes to the flow estimation accuracy. As a baseline, we first consider an asymmetric version of our model, which only relies on the data term and the pairwise term (Asymm). Then, we extend it to the symmetric case by estimating flow bi-directionally through enabling the forward-backward consistency term (Symm+c) and the occlusion-disocclusion symmetry term (Symm+s) separately. The full model has both these terms enabled (Symm+cs). We conduct this ablation study on the KITTI Optical Flow 2015 training set.

As shown in Table 4.3, the occlusion-disocclusion symmetry term has the most significant contribution to the accuracy of flow estimation. The error decreases substantially by about 20%. This observation follows one of our motivations that the accurate localization of occluded areas contributes to estimating flow more accurately.

	Non-occluded pixels			All pixels		
Method	Fl-bg	Fl-fg	Fl-all	Fl-bg	Fl-fg	Fl-all
Symm+cs	**6.52%**	**11.72%**	**7.41%**	**9.26%**	**13.94%**	**9.98%**
Symm+s	6.73%	11.90%	7.62%	9.49%	14.04%	10.19%
Symm+c	10.41%	18.17%	11.74%	13.72%	20.40%	14.74%
Asymm	8.39%	14.97%	9.51%	11.82%	17.41%	12.68%

Table 4.3: **Ablation study for each term** on KITTI 2015 training: The forward-backward consistency term (c) and the occlusion-disocclusion symmetry term (s) both play important roles in our symmetric model. Their combination (cs) shows the best results.

The forward-backward consistency term boosts the quality of the results further, but only when the symmetry term is turned on. This is because the forward-backward consistency relies on accurate estimates of the occlusion regions, which are only available when the occlusion-disocclusion symmetry is considered as well. When both terms are active we achieve the best accuracy, which highlights the benefit of our full symmetric pipeline.

4.4 DISCUSSION

In this chapter, we have proposed a symmetric optical flow method that jointly estimates optical flow in both directions and occlusion maps for each view by exploiting the symmetry properties possessed in the two consecutive images. We exploit both forward-backward consistency of the flow as well as occlusion-disocclusion symmetry, and formulate a piecewise rigid model. Our results on widely-used public optical flow benchmarks clearly demonstrate that our joint, symmetric approach yields significant improvements in flow estimation accuracy, especially in occluded areas. For the challenging KITTI benchmark, we report leading results even without employing any semantic knowledge or learning of appearance descriptors. We believe that a superpixel refinement, employing non-rigidity on top of our rigid motion model, or utilizing learned appearance descriptors will lead to further improvements.

5

ITERATIVE RESIDUAL REFINEMENT FOR JOINT OPTICAL FLOW AND OCCLUSION ESTIMATION

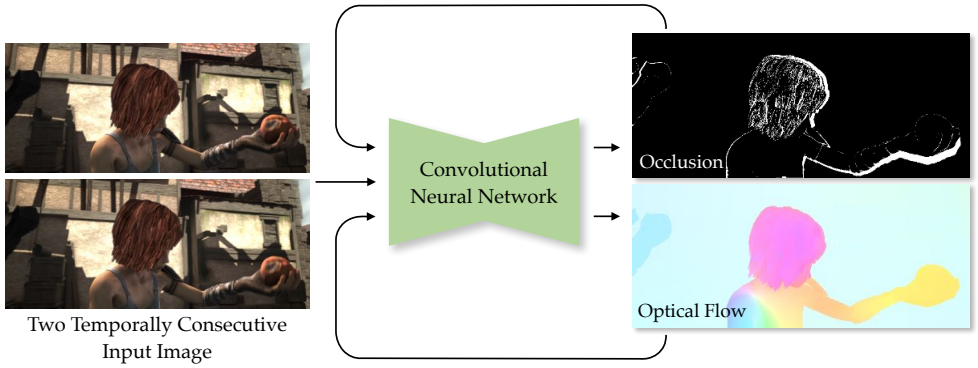


Figure 5.1: **Iterative Residual Refinement for Joint Optical Flow and Occlusion Estimation**: This chapter proposes an Iterative Residual Refinement (IRR) scheme for joint optical flow and occlusion estimation based on weight sharing.

CONTENTS

As discussed in Section 2.1.2.2, deep learning approaches to optical flow estimation have made rapid progress and led to a paradigm shift from traditional energy-based approaches to CNN-based approaches, demonstrating better accuracy and substantially faster run-time. To achieve high accuracy, many networks usually refine an initial flow estimate either through multiple stages or across the levels of a coarse-to-fine representation; however, this approach leads an increase of the number of network parameters.

With lessons learned from both traditional energy minimization approaches as well as residual networks, in this chapter we propose an *Iterative Residual Refinement* (IRR) scheme based on *weight sharing* that can be combined with several backbone networks. The IRR scheme reduces the number of parameters, improves the accuracy, or even achieves both. Furthermore, as our second joint objective, we show that integrating occlusion prediction and bi-directional flow estimation into our IRR scheme can further boost the accuracy. This highlights that optical flow and occlusion estimation can successfully exploits their relationship in CNN as well, as we demonstrate with the traditional energy-based model in Chapter 4. At the time of publication, our full network achieved the state-of-the-art results for both optical flow and occlusion estimation across several standard datasets.

5.1 INTRODUCTION

Despite that deep learning has lead a significant impact on optical flow estimation, as in many areas of computer vision (*e. g.*, object detection (Girshick et al., 2016) or human pose estimation (Tompson et al., 2014)), the accuracy of deep learning-based flow methods on public benchmarks (Butler et al., 2012; Menze et al., 2015b; 2018) had initially not surpassed that of classical approaches. Still, the efficient test-time inference has led to the widespread adoption of CNN-based optical flow methods as a sub-module in applications required to process temporal information, such as video object segmentation (Cheng et al., 2017), video recognition (Gadde et al., 2017; Nilsson and Sminchisescu, 2018; Zhu et al., 2017b), and video style transfer (Chen et al., 2017a).

FlowNet (Dosovitskiy et al., 2015) pioneered the use of CNNs in estimating optical flow and relied on a – by now standard – encoder-decoder architecture with skip connections, similar to semantic segmentation (Long et al., 2015), among others. Yet, because the flow accuracy remained behind that of traditional methods based on energy minimization, later work has focused on designing more powerful CNN architectures for optical flow. FlowNet2 (Ilg et al., 2017) resolved the accuracy limitations of FlowNet and started to outperform traditional approaches. Its main principle is to stack multiple FlowNet-family networks (Dosovitskiy et al., 2015), such that later stages effectively refine the output from the previous ones. However, one of the side effects of this stacking is the linearly increasing number of parameters, challenging the adoption in other applications. Also, stacked networks require sequential stage training rather than joint, resulting in a complex training procedure in practice.

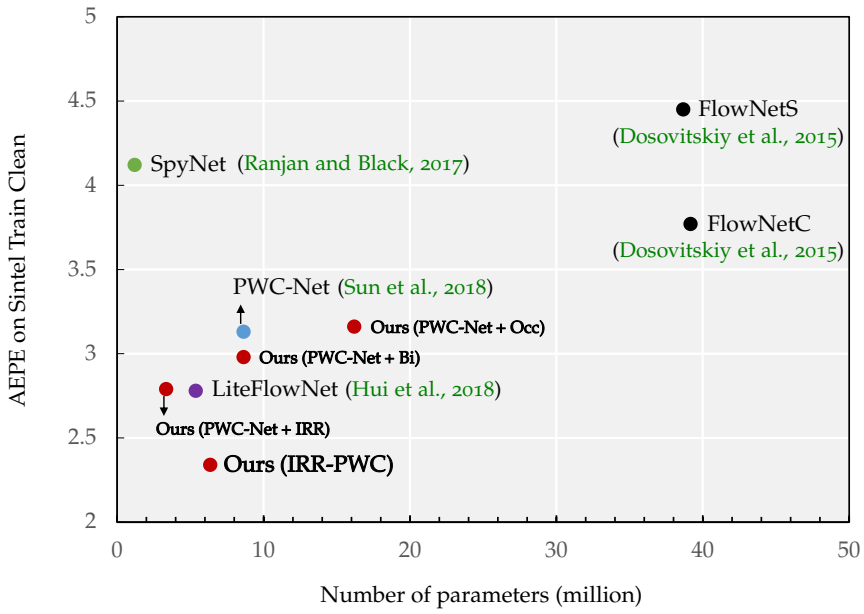
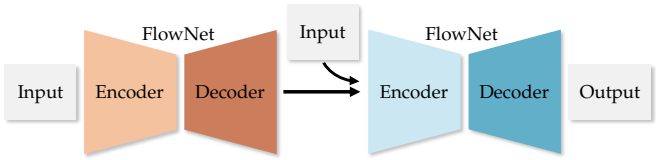


Figure 5.2: **Accuracy / network size tradeoff of CNNs for optical flow:** Combining our IRR (Iterative Residual Refinement), as well as bi-directional (Bi) and occlusion estimation (Occ) with PWC-Net (Sun et al., 2018) in comparison to previous work. Our full model (IRR-PWC), combining all three components, yields significant accuracy gains over Sun et al. (2018) while having many fewer parameters.

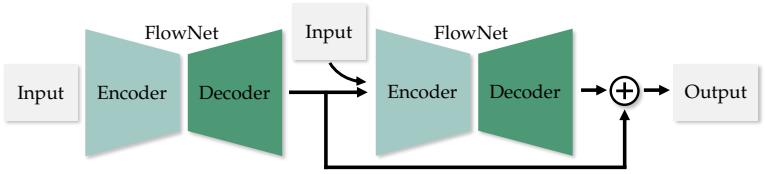
More recently, SpyNet (Ranjan and Black, 2017), PWC-Net (Sun et al., 2018), and LiteFlowNet (Hui et al., 2018) proposed lightweight networks that still achieve competitive accuracy (*cf*. Fig. 5.2). SpyNet adopts coarse-to-fine estimation in the network design, a well-known principle in classical approaches. It residually updates the flow across multiple levels of a spatial pyramid with individual trainable weights and demonstrates better accuracy than FlowNet even with far fewer model parameters. LiteFlowNet and PWC-Net further combine the coarse-to-fine strategy with multiple ideas from both classical methods and recent deep learning approaches. Particularly, PWC-Net outperformed all published methods on the common public benchmarks (Butler et al., 2012; Menze et al., 2015b; 2018).

Interestingly, many recent deep learning approaches for flow (Hui et al., 2018; Ilg et al., 2017; Ranjan and Black, 2017; Sun et al., 2018) have a common structure; that is, from a rough first flow estimate, later modules or networks refine the previous estimates across pyramid levels or through multiple chained networks. As each module assumes a particular functionality at the respective spatial resolution or is conditioned on the output of the preceding modules, the later modules or networks have their own trainable weights as illustrated in Fig. 5.3a. The downside is that this significantly increases the number of required model parameters. Please refer to Section 2.1.2.2 for a more in-depth review of the end-to-end regression architectures.

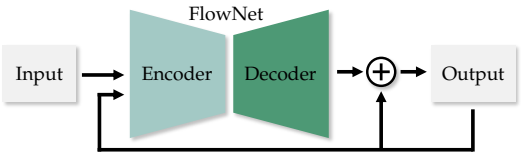
In this chapter, we take the lessons learned from traditional energy minimization-based optical flow approaches and proceed several steps further. Energy-based methods estimate the flow iteratively based on a consistent underlying energy with a



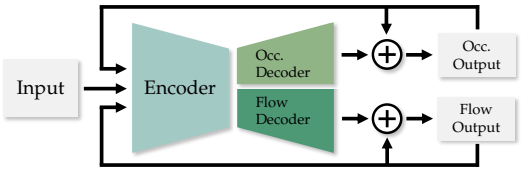
(a) FlowNet2-like (Ilg et al., 2017) stack of FlowNet networks.



(b) Iterative residual refinement version of *(a)*.



(c) A rolled version of *(b)*.



(d) Joint flow and occlusion estimation.

Figure 5.3: **From a standard network stack to our iterative residual refinement scheme with joint optical flow and occlusion estimation:** The stacked version of FlowNet *(a)* can be converted into an iterative residual refinement model *(b)* with the half number of parameters. Note that modules with the same color share their weights. We can re-interpret *(b)* as a rolled version *(c)*, making it more immediate to include occlusion estimation *(d)* for further accuracy improvement.

single set of parameters (Black and Anandan, 1996; Brox et al., 2004; Sun et al., 2014b). Therefore, our first research question is: *Can we iteratively refine flow with a deep network based on a single, shared set of weights?* Also, energy-based methods have benefited from bi-directional estimation and occlusion reasoning (Alvarez et al., 2007; Hur and Roth, 2017; Sun et al., 2014a; Xiao et al., 2006). Then, our second research question is: *Can deep learning approaches to optical flow similarly benefit from bi-directional estimation with occlusion reasoning?*

We address these questions and make a number of contributions: *(i)* We first propose an Iterative Residual Refinement (IRR) scheme that takes the output from a previous iteration as input and refines it iteratively by only using a *single* network block with *shared weights*. *(ii)* We demonstrate the applicability to two popular networks, FlowNet (Dosovitskiy et al., 2015) (Fig. 5.3c) and PWC-Net (Sun et al., 2018) (Fig. 5.5). For FlowNet, we can significantly increase the accuracy without adding parameters; for PWC-Net, we can reduce the number of parameters while even improving the accuracy (Fig. 5.2). *(iii)* Next, we demonstrate the integration with occlusion estimation (Fig. 5.3d) as our second joint objective. *(iv)* We further extend the scheme to bi-directional flow estimation, which turns out to be only beneficial when combined with occlusion estimation. Unlike previous work (Ilg et al., 2018), our scheme enables the flow accuracy to benefit from joint occlusion estimation. *(v)* We finally propose

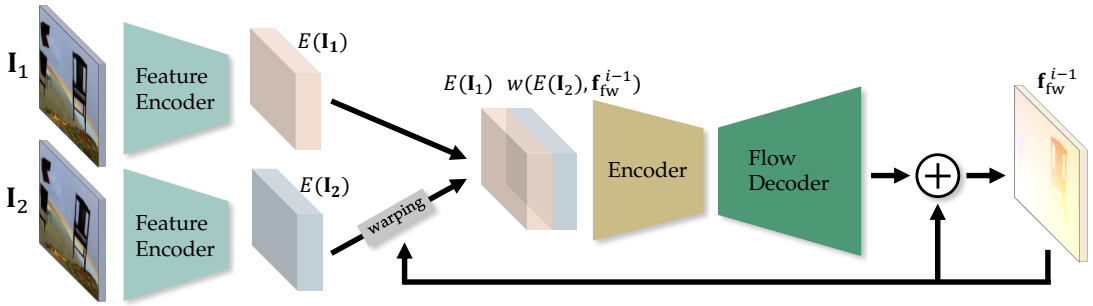


Figure 5.4: **Our IRR version of FlowNetS** (Dosovitskiy et al., 2015)**.** The model iteratively estimates residual flow from the previous output. Note that we apply warping after several encoder layers, see text for details.

lightweight bilateral filtering and occlusion upsampling layers for refined motion and occlusion boundaries.

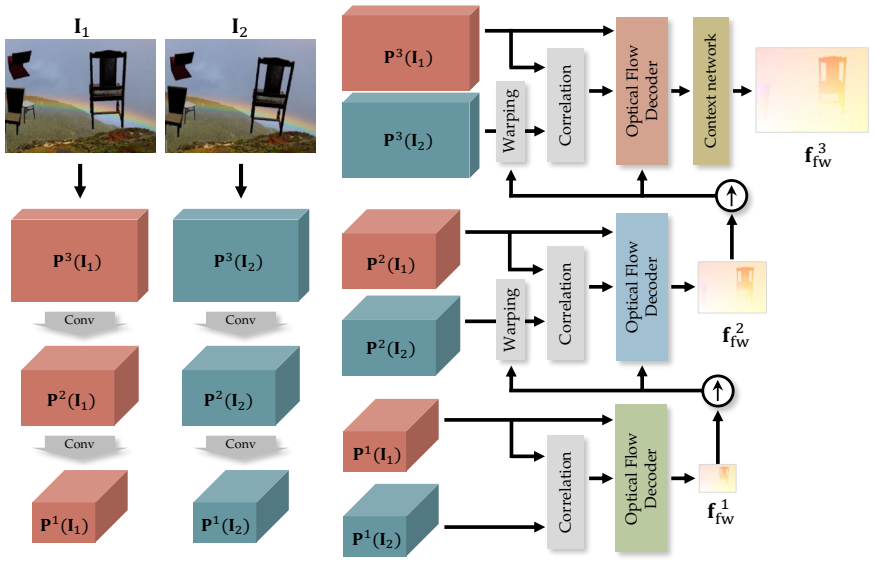
Application of our proposed scheme to the two backbone networks, FlowNet and PWC-Net, yields significant improvements in flow accuracy by 18.5% and 17.7%, respectively, across multiple datasets. In case of PWC-Net, we achieve this accuracy gain using 26.4% *fewer* parameters. Note that occlusion estimation and bi-directional flow are additional outcomes as by-products of this improvement.

5.2 ITERATIVE RESIDUAL REFINEMENT

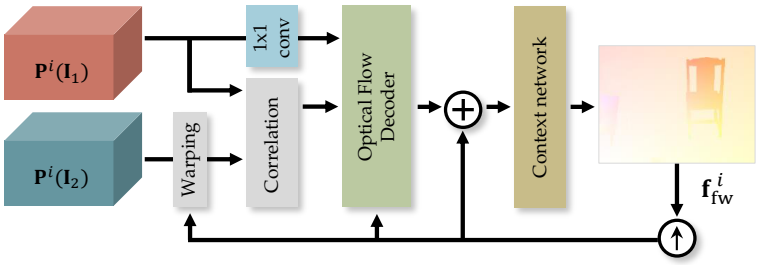
5.2.1 *Core concepts & base networks*

IRR (Iterative Residual Refinement) with shared weights. The basic problem setup is to estimate (forward) optical flow $\mathbf{f}_{\text{fw}}$ from the reference frame $\mathbf{I}_1$ to the target frame $\mathbf{I}_2$. The main concept of our IRR scheme is to make a model learn to residually refine its previous estimate by iteratively *re-using* the same network block with *shared weights*. We pursue two scenarios: *(i)* We increase the accuracy without adding parameters or complicating the training procedure, by iteratively re-using a single network to keep refining its previous estimate; or *(ii)* we aim toward a more compact model by substituting multiple network blocks assuming the same basic functionality with only a single block.

IRR with FlowNet. Addressing the first scenario, we propose an iterative residual refinement version of FlowNetS (Dosovitskiy et al., 2015), *cf*. Fig. 5.4 for an overview. Our IRR version iteratively estimates residual flow with multiple iterations using one single FlowNetS; the final result is the sum of residual flows from all iteration steps. We use one shared encoder E for feature extraction from each input image $\mathbf{I}_1$ and $\mathbf{I}_2$, similar to FlowNetC, and concatenate the two feature maps after warping the second feature map based on the estimated flow $\mathbf{f}_{\text{fw}}^{i-1}$ from the previous iteration $i-1$. Then



(a) Original PWC-Net (Sun et al., 2018).



(b) Our IRR version of PWC-Net.

Figure 5.5: **Our IRR version of PWC-Net**, which uses only one single shared decoder over the pyramid levels, see text for details.

we input the concatenated feature maps to the decoder D to estimate the residual (forward) flow at iteration i:

$$\mathbf{f}_{\text{fw}}^{i} = D\Big(E(\mathbf{I}_1), w\big(E(\mathbf{I}_2), \mathbf{f}_{\text{fw}}^{i-1}\big)\Big) + \mathbf{f}_{\text{fw}}^{i-1}, \tag{5.1}$$

where $w(\cdot,\cdot)$ is a bilinear interpolation function for backward warping (Jaderberg et al., 2015). Here, warping the second feature map is crucial as it yields a suitable input for estimating the appropriate *residual* flow. This yields much improved accuracy while re-using the same network with only slight modifications and not requiring additional training stages.

IRR with PWC-Net. Based on the classical coarse-to-fine principle, PWC-Net (Sun et al., 2018) and SpyNet (Ranjan and Black, 2017) both use multiple repetitive modules for the same purpose but with *separate weights*. Fig. 5.5a shows a 3-level PWC-Net (for ease of visualization, originally 7-level) that incrementally updates the estimation across the pyramid levels with individual decoders for each level. Adopting our IRR scheme here to address the second scenario, we can substitute the multiple decoders

with only one shared decoder that iteratively refines the output over all the pyramid levels, *cf*. Fig. 5.5b. We set the number of iterations equal to the number of pyramid levels, keeping the original pipeline but with fewer parameters and a more compact representation:

$$\mathbf{f}_{\text{fw}}^{i} = D\Big(\mathbf{P}^{i}(\mathbf{I}_1), c\big(\mathbf{P}^{i}(\mathbf{I}_1), w\big(\mathbf{P}^{i}(\mathbf{I}_2), \hat{\mathbf{f}}_{\text{fw}}^{i-1}\big)\big), \hat{\mathbf{f}}_{\text{fw}}^{i-1}\Big) + \hat{\mathbf{f}}_{\text{fw}}^{i-1} \tag{5.2a}$$

with

$$\hat{\mathbf{f}}_{\text{fw}}^{i-1} = 2 \cdot \uparrow(\mathbf{f}_{\text{fw}}^{i-1}), \tag{5.2b}$$

where $\mathbf{P}^{i}$ is the feature map at pyramid level i, $c(\cdot,\cdot)$ calculates a cost volume, and $\uparrow$ performs $2\times$ bilinear upsampling to twice the resolution of the previous flow field. As the dimension increases, we also scale the flow magnitude accordingly (Eq. 5.2b).

One important change from the original PWC-Net (Sun et al., 2018), which estimates flow for each level on the original scale, is that we estimate flow for each level at its native spatial resolution. This enables us to use only one shared decoder and yet make it possible to handle different resolutions across all levels. When calculating the loss, we revert back to the original scale to use the same loss function.

In addition, we add a 1×1 convolution layer after the input feature map $\mathbf{P}^{i}(\mathbf{I}_1)$ to make the number of feature maps input to the decoder D be equal across the pyramid levels. This enables us to use one single shared decoder with a fixed number of input channels across the pyramid.

By the way, as a follow-up approach of our method, RAFT (Teed and Deng, 2020) proposes to recurrently and residually update optical flow using a GRU module and a pre-computed multli-scale 4D cost volume. In Appendix B.3, we provide a comparison on technical designs of both methods.

Occlusion estimation. It is widely reported that jointly localizing occlusions and estimating optical flow can benefit each other (Alvarez et al., 2007; Ballester et al., 2012; Hur and Roth, 2017; Ince and Konrad, 2008; Sun et al., 2014a; Unger et al., 2012; Xiao et al., 2006). Toward leveraging this in the setting of CNNs, we attach an additional decoder estimating occlusion $\mathbf{o}_1^i$ in the first frame at the end of the encoder, in parallel to the flow decoder as shown in Fig. 5.3d, similar to Janai et al. (2018) and Neoral et al. (2018). The occlusion decoder has the same configuration as the flow decoder, but the number of output channels is 1 (instead of 2 for flow). The input to the occlusion decoder is the same as to the flow decoder.

5.2.2 *Joint optical flow and occlusion estimation*

Iteratively re-using residual subnetworks and adding occlusion decoders are independent, easily combined together, and adaptable to many types of optical flow base networks. Beyond simply adopting these two concepts, we additionally propose several ideas to improve the accuracy further in a joint estimation setup: *(i)* bi-directional estimation, *(ii)* bilateral refinement of flow and occlusion, and *(iii)* an occlusion upsampling layer.

Bi-directional estimation. Based on the basic IRR setup for joint flow and occlusion estimation in Fig. 5.3d, we first perform bi-directional flow and occlusion estimation by simply switching the order of the input feature maps for the decoder (Meister et al., 2018; Wang et al., 2018). This yields backward flow $\mathbf{f}^i_{\text{bw}}$ and occlusion $\mathbf{o}^i_2$ in the second frame. Note that bi-directional estimation requires no extra convolutional weights as it again re-uses the same shared decoders. As we shall see, estimating both forward and backward flow together yields at most minor accuracy improvements itself, but we find that exploiting forward-backward consistency is crucial for estimating more accurate occlusions.

Bilateral refinement of flow and occlusion. Blurry estimates, particularly near motion boundaries, have recently been identified as a main limitation of standard optical flow decoders in CNNs. To address this, bilateral filters or local attention maps (Harley et al., 2017; Hui et al., 2018) have been proposed as viable solutions. We also adopt this idea in our setup, extend it to refine optical flow *and* occlusion using bilateral filters, but with *weight sharing* across all iteration steps.

Similar to Hui et al. (2018), we construct learned bilateral filters individualized to each pixel and apply them to each flow component u, v and the occlusion separately:

$$\tilde{f}^i_{\text{fw},u}(x,y) = g_{\text{fw}}(x,y) * f^i_{\text{fw},u}(x,y) \tag{5.3a}$$

$$\tilde{f}^i_{\text{fw},v}(x,y) = g_{\text{fw}}(x,y) * f^i_{\text{fw},v}(x,y) \tag{5.3b}$$

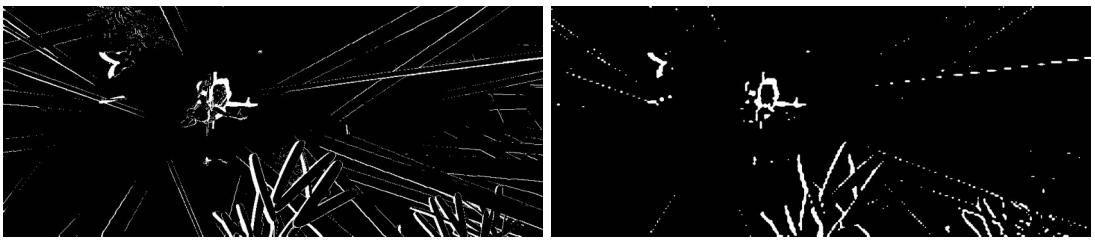
$$\tilde{o}^i_1(x,y) = g_o(x,y) * o^i_1(x,y), \tag{5.3c}$$

where, *e.g.*, $\tilde{f}^i_{\text{fw},u}(x,y)$ is the filtered horizontal flow at (x,y), $g_{\text{fw}}(x,y)$ is the $w \times w$ learned bilateral filter kernel for flow at (x,y), and $f^i_{\text{fw},u}(x,y)$ is the $w \times w$ patch of the horizontal flow centered at (x,y). Note that we construct the kernels for flow and occlusion separately as motion and occlusion boundaries are not necessarily aligned.

For constructing the bilateral filter for the flow, we follow the strategy of Hui et al. (2018), and for occlusion we input occlusion estimates, a feature map, and a warped feature map from the other temporal direction. One important difference to Hui et al. (2018) is that we do not need separate learnable convolutional weights for every iteration step or every pyramid level. Our IRR design enables re-using the same weights for constructing the bilateral filters for all iteration steps or pyramid levels. In case of adapting to PWC-Net, our bilateral refinement adds only 0.69M parameters, which is 2.4× less than the scheme of Hui et al. (2018), adding 1.66M parameters.

Occlusion upsampling layer. One common trait of FlowNet (Dosovitskiy et al., 2015) and PWC-Net (Sun et al., 2018) is that the output resolution of flow from the CNN is a quarter ($\frac{1}{4}H \times \frac{1}{4}W$) of the input resolution ($H \times W$), which is then bilinearly upscaled to the input resolution. The reasons for not directly estimating at full resolution are a marginal accuracy improvement and the GPU computation and memory overhead.

Yet for estimating occlusion, there is a significant accuracy loss when estimating only at a quarter resolution. On the Sintel dataset, we conduct an oracle study by downscaling the ground-truth occlusion maps to a quarter size and then upscaling them back. The F-score of the reconstructed occlusion maps was 0.777, suggesting



(a) Ground-truth occlusion map.

(b) Reconstructed occlusion map from a quarter resolution of *(a)*.

Figure 5.6: **Oracle study** showing the limitation of outputting low-resolution occlusion maps.

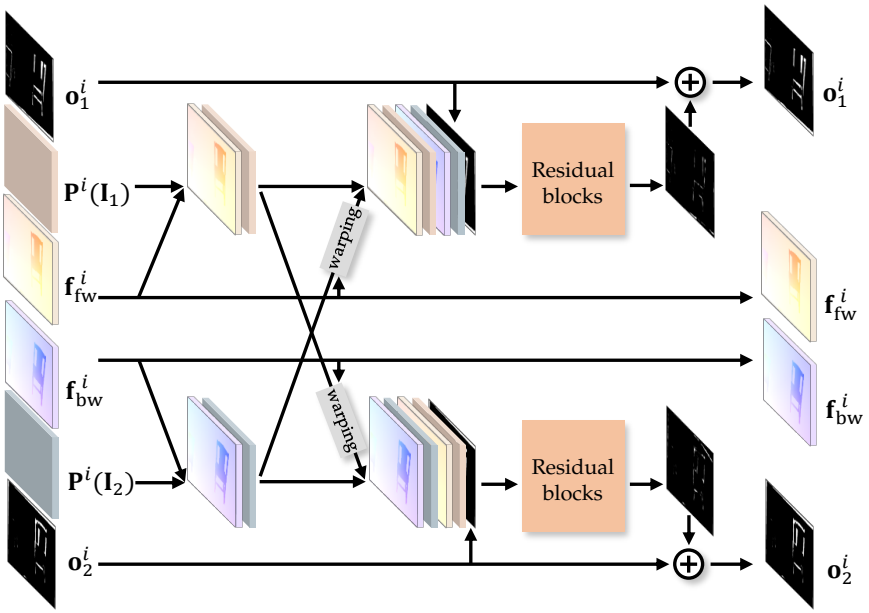


Figure 5.7: **Occlusion upsampling layer**: Inputs are bi-linearly upsampled flow and upsampled occlusion using nearest neighbor. Residual blocks then improve the occlusion accuracy using residual occlusion updates at the full output resolution.

a significant accuracy limitation. As seen in Fig. 5.6, quarter-resolution occlusion maps cannot really represent fine occlusions, through which the major loss in F-score occurs. This strongly emphasizes the importance of estimating at full resolution.

To estimate more accurate occlusion at higher resolution, we attach an upsampling layer at the end of network to upscale optical flow and occlusion together back to the input resolution. Fig. 5.7 illustrates our upsampling layer. For optical flow, we found bilinear upsampling to be sufficient. For occlusion, we first perform nearest-neighbor upsampling, which is fed into a CNN module to estimate the residual occlusion on the upsampled occlusion map. The CNN module consists of three residual blocks (Lim et al., 2017), which receive flow, a feature map from the encoder, warped flow, and a warped feature map from the other temporal direction. Putting the warped feature map and flow from the other direction enables exploiting the classical forward-

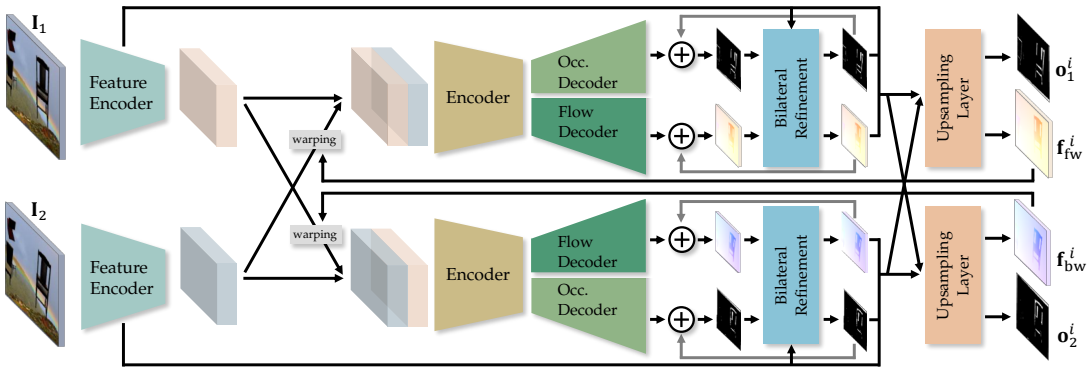


Figure 5.8: **Joint optical flow and occlusion estimation: bi-directional estimation, bilateral refinement, and upsampling layer** (in the FlowNet setting): We estimate flow in both temporal directions and occlusion maps in both frames by switching the order of inputs to the decoder. Bilateral refinement and the upsampling layer further improve the accuracy of flow and occlusion. Modules with the same color share their weights.

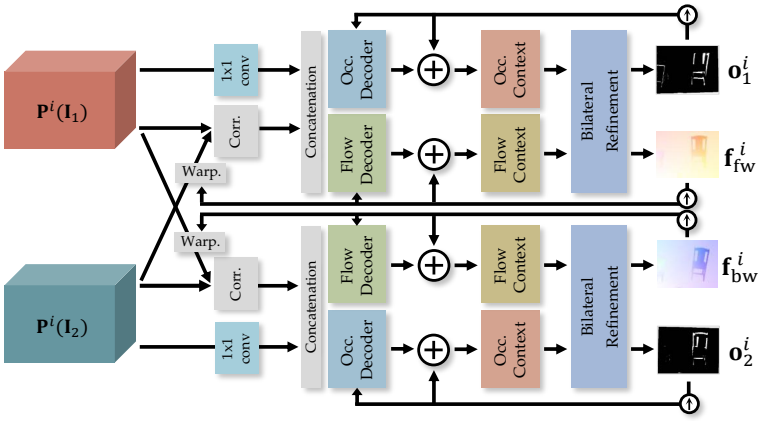
backward consistency for estimating the occlusion. We provide further details in the appendix, Appendix B.1.

Full model. Fig. 5.8 and Fig. 5.9 visualize our final variants of FlowNet and PWC-Net respectively, enabling joint optical flow and occlusion estimation based on bi-directional estimation, bilateral refinement, and the occlusion upsampling layer. The PWC-Net variant (we name it as IRR-PWC) in Fig. 5.9 first iteratively and residually estimates optical flow and occlusion up to a quarter resolution of the input image (*cf.*, Fig. 5.9a), given a 7-level feature pyramid as in the original PWC-Net (Sun et al., 2018). Then, given the estimates at the 5^{th} level, we show how we use our occlusion upsampling layer in Fig. 5.9b to scale the estimates up to the original resolution. The upsampling layer, illustrated in Fig. 5.7, upscales the resolution by $2\times$ at once, and applying the upsampling layer at the 6^{th} and 7^{th} level scales the quarter resolution estimate back to the original resolution.

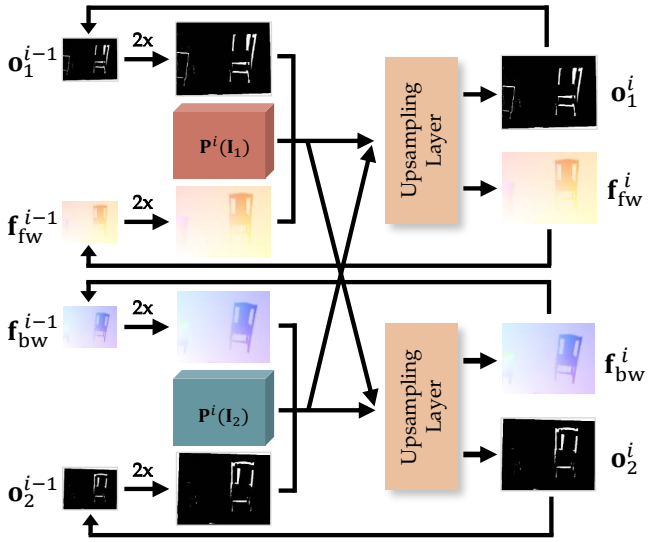
Initialization. To bootstrap our iterative estimation, we input zero as initial optical flow (*i. e.* $\mathbf{f}^0_{\text{fw}}$ and $\mathbf{f}^0_{\text{bw}}$) and occlusion (*i. e.* $\mathbf{o}^0_1$ and $\mathbf{o}^0_2$) into the first stage. Note that 0 indicates non-occluded (visible) and 1 indicates occluded.

Training loss. Let N be the total number of steps in our iterative setting. Then we predict a set of forward optical flow maps $\mathbf{f}^i_{\text{fw}}$, backward optical flow $\mathbf{f}^i_{\text{bw}}$, occlusion maps in the first image $\mathbf{o}^i_1$ and in the second image $\mathbf{o}^i_2$ for each iteration step, where $i = 1, \ldots, N$. Forward and backward optical flow are supervised using the $\mathcal{L}_{2,1}$ norm as

$$l^i_{\text{flow}} = \frac{1}{2} \sum \left(\|\mathbf{f}^i_{\text{fw}} - \mathbf{f}_{\text{fw,GT}}\|_2 + \|\mathbf{f}^i_{\text{bw}} - \mathbf{f}_{\text{bw,GT}}\|_2 \right), \tag{5.4}$$



(a) Jointly estimating optical flow and occlusion up to a quarter resolution of the original input, *i. e.*, pyramid levels $1 \leq i \leq 5$.



(b) Upsampling optical flow and occlusion using the upsampling layer, *i. e.*, pyramid levels $6 \leq i \leq 7$.

Figure 5.9: **IRR-PWC**: Our PWC-Net variant with joint optical flow and occlusion estimation based on bi-directional estimation, bilateral refinement, and the occlusion upsampling layer. *(a)* Our IRR-PWC model jointly estimates optical flow and occlusion up to a quarter resolution of the input image (*i. e.*, up to the 5^{th} level), the same as the original PWC-Net. *(b)* Then, we use our occlusion upsampling layer to upscale the outputs back to the original resolution while improving accuracy.

whereas for the supervision of the two occlusion maps we use a weighted binary cross-entropy

$$\begin{aligned} l^i_{\text{occ}} = -\tfrac{1}{2} \sum \big(& w^i_1 o^i_1 \log o_{1,\text{GT}} + \bar{w}^i_1 (1-o^i_1) \log(1-o_{1,\text{GT}}) \\ & + w^i_2 o^i_2 \log o_{2,\text{GT}} + \bar{w}^i_2 (1-o^i_2) \log(1-o_{2,\text{GT}}) \big). \end{aligned} \tag{5.5}$$

Here, we apply the weights $w^i_1 = \frac{H \cdot W}{\sum o^i_1 + \sum o_{1,\text{GT}}}$ and $\bar{w}^i_1 = \frac{H \cdot W}{\sum (1-o^i_1) + \sum (1-o_{1,\text{GT}})}$ to take into account the number of predictions and true labels.

Our final loss is the weighted sum of the two losses above, taken over all iteration steps using the same multi-scale weights α_s as in the original papers. In case of FlowNet (Dosovitskiy et al., 2015), the final loss becomes

$$l_{\text{FlowNet}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{s=s_0}^{S} \alpha_s (l^{i,s}_{\text{flow}} + \lambda \cdot l^{i,s}_{\text{occ}}), \tag{5.6}$$

where s denotes the scale index given in Fig. 3 of Dosovitskiy et al. (2015). In case of PWC-Net (Sun et al., 2018), the number of scales is equal to the number of iterations, hence the final loss is

$$l_{\text{PWC-Net}} = \frac{1}{N} \sum_{i=1}^{N} \alpha_i (l^i_{\text{flow}} + \lambda \cdot l^i_{\text{occ}}). \tag{5.7}$$

λ weighs the flow against the occlusion loss. In every iteration, we calculate the λ that makes the loss of the flow and the occlusion be equal. We empirically found that this strategy yields better accuracy than just using a fixed trade-off.

5.3 EXPERIMENTS

5.3.1 *FlyingChairsOcc dataset*

Lacking a suitable dataset, we create our own dataset for the supervision of bi-directional flow and the two occlusion maps, with ground truth for forward flow, backward flow, and occlusion maps at the first and second frame. To build the dataset, we follow the exact protocol of the FlyingChairs dataset (Dosovitskiy et al., 2015). We refer to this dataset as *FlyingChairsOcc*. Figure 5.10 shows some examplar images from the dataset.

We crawl 964 background images with a resolution of 1024×768 from Flickr and Google using the keywords *cityscape*, *street*, and *mountain*. As foreground objects, we use 809 chair images rendered from CAD models with varying views and angles (Aubry et al., 2014). Then we follow the exact protocol of (Dosovitskiy et al., 2015) for generating image pairs, including the number of foreground objects, object size, and random parameters for generating the motion of each object. As the motion is parametrized by a 3×3 matrix, it is easy to calculate not only backward ground-truth flow but also occlusion maps by conducting visibility checks. The number of images in the training and validation sets are the same as in FlyingChairs (*i. e.*, 22 232 and 640, respectively).

Figure 5.10: **FlyingChairsOcc dataset**: *(left to right)* the first image, the second image, forward optical flow, backward optical flow, occlusion map at the first image, occlusion map at the second image.

5.3.2 *Implementation details*

TRAINING DETAILS. We follow the training settings of FlowNet respective PWC-Net for a fair comparison. We use the same geometric and photometric augmentations with additive Gaussian noise as described in Ilg et al. (2017). After applying the geometric augmentation on the occlusion ground truth, we additionally check for pixels moving outside of the image boundary (*i. e.*, out-of-bound pixels) and set them as occluded. Note that no multi-stage training is needed.

We first train the proposed model on our FlyingChairsOcc dataset with learning rate schedule S_{short} (instead of S_{long}), described in Ilg et al. (2017). Next, we fine-tune on the FlyingThings3D-subset dataset (Mayer et al., 2016), which contains much larger displacements; we use half the S_{fine} learning rate schedule (Ilg et al., 2017). We empirically found that using shorter schedules was enough as our model converged faster. We finally fine-tune on different public benchmark datasets, including Sintel (Butler et al., 2012) and KITTI (Menze et al., 2015b; 2018), following the fine-tuning protocol of (Sun et al., 2020). We use a smaller minibatch size of 4, as our model implicitly increases the batch size by performing iterative bi-directional estimation with a single model.

Lacking other ground truth, we only use the forward flow and the occlusion map for the first frame for supervision on Sintel; for KITTI we only use the forward flow. Importantly, our model is still trainable when ground truth is available only for one direction (*e. g.*, forward flow with occlusion map at the first frame), since both temporal directions share the same "unidirectional" decoder.

5.3.3 *Ablation study*

To see the effectiveness of each proposed component, we conduct an ablation study by training our model in multiple settings. All models are trained on the FlyingChairsOcc dataset with the S_{short} schedule and tested on multiple datasets to assess generalization across datasets. We use a minibatch size of 4 when either bi-directional estimation or iterative residual refinement is on, or the original minibatch size of 8, otherwise. For

	Bi	Occ	IRR	Chairs Full	ChairsOcc Validation	Sintel Clean Training	Sintel Final Training	Rel. Param.
FlowNet (Dosovitskiy et al., 2015)				2.39	2.27	4.35	5.44	0 %
	✓			2.43	2.30	4.40	5.53	0 %
		✓		2.29	2.18 (0.690)	4.26 (0.521)	5.51 (0.493)	+38.5%
			✓	2.36	2.22	3.77	5.00	0 %
	✓	✓		2.31	2.20 (0.691)	4.21 (0.515)	5.46 (0.488)	+38.5%
	✓		✓	2.14	2.00	3.45	4.96	0 %
		✓	✓	2.22	2.10 (0.689)	3.56 (0.507)	5.03 (0.486)	+38.5%
	✓	✓	✓	2.05	1.91 (0.699)	3.40 (0.528)	5.08 (0.502)	+38.5%
	✓	✓	✓+	**1.92**	**1.77 (0.736)**	**3.32 (0.596)**	**4.92 (0.560)**	+40.7%
PWC-Net (Sun et al., 2018)				2.03	1.89	3.13	4.41	0 %
	✓			2.06	1.87	2.98	4.14	0 %
		✓		1.94	1.79 (0.706)	3.16 (0.616)	4.35 (0.581)	+87.4%
			✓	2.01	1.83	2.79	4.10	−61.2%
	✓	✓		1.99	1.82 (0.696)	3.01 (0.618)	4.39 (0.581)	+87.4%
	✓		✓	2.08	1.90	2.80	4.13	−61.2%
		✓	✓	1.91	1.73 (0.700)	2.64 (0.630)	4.09 (0.593)	−34.7%
	✓	✓	✓	1.98	1.81 (0.698)	2.69 (0.633)	4.03 (0.598)	−34.7%
	✓	✓	✓+	**1.67**	**1.48 (0.757)**	**2.34 (0.677)**	**3.95 (0.624)**	−26.4%

Table 5.1: **Ablation study of our design choices on the two baseline models.** The numbers indicate the Average End-Point Error (AEPE) for optical flow (the lower the better) and the average F1-score for occlusion in parentheses, where available (the higher the better). **Bi**: Bi-directional estimation, **Occ**: Joint occlusion estimation, **IRR**: Iterative residual refinement, **IRR+**: Iterative residual refinement including bilateral refinement and occlusion upsampling layer. The final column reports the relative changes on the number of parameters comparing to the vanilla baseline.

a simpler ablation study, we use two iteration steps when applying IRR on FlowNet (Dosovitskiy et al., 2015).

Table 5.1 assesses the optical flow in terms of the Average End-Point Error (AEPE) and occlusion estimation with the average F1-score, if applicable for the respective configuration. In contrast to findings in recent work (Ilg et al., 2018), estimating occlusion together yields a gradual improvement of the flow of up to 5% on the training domain, and an even bigger improvement across different datasets when combined on top of bi-directional estimation (Bi) or IRR. We believe this to mainly stem from using a separate occlusion decoder instead of a joint decoder (Ilg et al., 2018). Bi-directional estimation by itself yields at most a marginal improvement on flow, but it is important for the input of the occlusion upsampling layer, which brings very large benefits on occlusion estimation. Iterative residual refinement yields consistent improvements in flow accuracy on the training domain, and perhaps surprisingly *a much better generalization across datasets*, with up to 10% improvement in EPE. We presume that this better generalization comes from training a single decoder to handle feature maps from all iteration steps or pyramid levels, which encourages generalization even across datasets. The benefits of using IRR become even clearer

Figure 5.11: **Qualitative examples from the ablation study on PWC-Net**: *(a)* Overlapped input images, *(b)* ground-truth flow, *(c)* the original PWC-Net (Sun et al., 2018), *(d)* our PWC-Net with IRR, *(e)* our PWC-Net with Bi-Occ-IRR, and *(f)* our full model (*i. e.*, IRR-PWC). More examples follow in Appendix B.2.2.

when combined with other components. For example, FlowNet with Bi, Occ, and IRR demonstrates up to 20% improvement in EPE on Sintel Clean compared to only using Bi and Occ. Additionally, the bilateral refinement and the upsampling layer significantly improve the accuracy of both flow and occlusion with a small overhead of only 0.83M parameters. For PWC-Net, we obtain a significant accuracy boost of 17.7% on average over the baseline, while reducing the number of parameters by 26.4%. We name the full versions of the models including all modules **IRR-FlowNet** and **IRR-PWC**. Fig. 5.11 highlights the improvement of the flow from our proposed components with qualitative examples. Please note the completeness and sharp boundaries.

Bilateral refinement. We compare our bilateral refinement layer with the refinement layer of LiteFlowNet (Hui et al., 2018) based on a PWC-Net with Bi, Occ, and IRR components enabled. Table 5.2 shows that the benefit of our design choice (*i. e.*, sharing weights) holds for bilateral refinement as well, yielding better accuracy for flow and particularly for occlusion, with 2.5× fewer parameters than that of Hui et al. (2018).

Occlusion upsampling layer. Similar to our upsampling layer, Ilg et al. (2018) use a refinement network from FlowNet2 (Ilg et al., 2017) to upsample the intermediate quarter-resolution outcome back to the original resolution. We compare our upsampling layer with the refinement network from Ilg et al. (2017, 2018), adding it to our network based on a PWC-Net backbone with Bi, Occ, IRR, and the bilateral refinement layer enabled. Table 5.3 shows the clear benefits of using our upsampling layer,

Method	Chairs Full	ChairsOcc Validation	Sintel Clean Training	Sintel Final Training	Rel. Param.
No refinement	1.98	1.81 (0.698)	2.69 (0.633)	4.03 (0.598)	0 %
Ours	**1.66**	**1.45 (0.735)**	**2.32 (0.648)**	3.90 **(0.602)**	+12.3%
LiteFlowNet's (Hui et al., 2018)	1.74	1.58 (0.688)	2.34 (0.596)	**3.86** (0.543)	+29.5%

Table 5.2: Comparison of our bilateral refinement layer against that of LiteFlowNet (Hui et al., 2018).

Method	Chairs Full	ChairsOcc Validation	Sintel Clean Training	Sintel Final Training	Rel. Param.
No upsampling	**1.66**	**1.45** (0.735)	**2.32** (0.648)	**3.90** (0.602)	0 %
Ours	1.67	1.48 **(0.757)**	2.34 **(0.677)**	3.95 **(0.624)**	+0.49%
Ilg et al. (2017, 2018)	2.18	2.01 (0.712)	2.90 (0.624)	4.37 (0.577)	+9.21%

Table 5.3: Comparison of our occlusion upsampling layer and the refinement network from FlowNet2 (Ilg et al., 2017; 2018).

yielding significant gains in both tasks while requiring fewer parameters. We provide qualitative examples in Appendix B.2.1. The refinement network from FlowNet2 (Ilg et al., 2017) actually degrades the accuracy of flow estimation. We presume this may stem from differences in training. FlowNet2's refinement layer may require piece-wise training, while our model is trained all at once.

Different IRR steps on FlowNet. For FlowNet, we can freely choose the number of IRR steps as we iteratively refine previous estimates by re-using a single network. We try different numbers of IRR steps on vanilla FlowNetS (Dosovitskiy et al., 2015) (*i. e.*, without Bi or Occ) and compare with stacking multiple FlowNetS networks. All networks are trained on FlyingChairsOcc with the S_{short} schedule, minibatch size of 8, and tested on Sintel Clean.

As shown in Table 5.4, the accuracy keeps improving with more IRR steps and stably settles at more than 4 steps. In contrast, stacking multiple FlowNetS networks overfits on the training data after 3 steps, and is consistently outperformed by IRR with the same number of stages. This clearly demonstrates the advantage of our IRR scheme over stacking: *better accuracy without linearly increasing the number of parameters*.

5.3.4 *Optical flow benchmarks*

We test the accuracy of our IRR-PWC on the public Sintel (Butler et al., 2012) and KITTI (Menze et al., 2015b; 2018) benchmarks. When fine-tuning, we use the robust training loss as in Hui et al. (2018) and Sun et al. (2018, 2020) for flow, and standard binary cross-entropy for occlusion. On Sintel Final in Table 5.5, our IRR-PWC achieves a new state of the art among 2-frame methods. Comparing to the PWC-Net baseline (*i. e.*, PWC-Net-ft-final) trained in the identical setting, our contributions improve the flow accuracy by 9.18% on Final and 12.36% on Clean, while using 26.4% fewer

Number of iterations or stacking stages	1	2	3	4	5
IRR on a single FlowNetS	**4.358**	**3.545**	**3.325**	**3.303**	**3.302**
Stacking multiple FlowNetS	4.445	3.553	3.377	3.391	3.517

Table 5.4: $n\times$ IRR *vs.* $n\times$ stacking: EPE on Sintel Clean

Method	Training		Test		Parameters
	Clean	Final	Clean	Final	
GMA† (Jiang et al., 2021)	(0.62)	(1.06)	**1.39**	**2.47**	5.9 M
Separable flow† (Zhang et al., 2021)	(0.69)	(1.10)	1.50	2.67	6.0 M
RAFT† (Teed and Deng, 2020)	(0.77)	(1.27)	1.61	2.86	5.3 M
RAFT-A† (Sun et al., 2021)	–	–	2.01	3.14	5.9 M
MaskFlowNet (Zhao et al., 2020)	–	–	2.52	4.17	–
ContinualFlow_ROB†§ (Neoral et al., 2018)	–	–	3.34	4.53	14.6 M
VCN (Yang and Ramanan, 2019)	(1.66)	(2.24)	2.81	4.40	6.2 M
MFF§ (Ren et al., 2019)	–	–	3.42	4.57	N/A
IRR-PWC (Ours)	(1.92)	(2.51)	3.84	4.58	6.36M
PWC-Net+† (Sun et al., 2020)	(1.71)	(2.34)	3.45	4.60	8.75M
ProFlow§(Maurer and Bruhn, 2018)	–	–	2.82	5.02	–
PWC-Net-ft-final (Sun et al., 2020)	(2.02)	(2.08)	4.39	5.04	8.75M
DCFlow (Xu et al., 2017)	–	–	3.54	5.12	–
FlowFieldsCNN (Bailer et al., 2017)	–	–	3.78	5.36	5.00M
MR-Flow (Wulff et al., 2017)	1.83	3.59	2.53	5.38	–
LiteFlowNet (Hui et al., 2018)	(1.35)	(1.78)	4.54	5.38	5.37M
S2F-IF (Yang and Soatto, 2017)	–	–	3.50	5.42	–
SfM-PM (Maurer et al., 2018)	–	–	2.91	5.47	–
FlowFields++ (Schuster et al., 2018a)	–	–	2.94	5.49	–
FlowNet2 (Ilg et al., 2017)	(2.02)	(3.14)	3.96	6.02	162.5 M

Table 5.5: MPI **Sintel Flow**: Average End-Point Error (AEPE) and number of CNN parameters. §using more than 2 frames, †using additional datasets for better accuracy.

parameters. On KITTI 2015 in Table 5.6, our IRR-PWC again outperforms all published 2-frame methods, improving over the baseline PWC-Net.

When fine-tuning on benchmarks, our important observations are that our model *(i)* converges much faster than the baseline and *(ii)* overfits to the training split less, demonstrating much better accuracy on the test set despite slightly higher error on training split. This highlights the benefit of our IRR scheme: better generalization *even on the training domain* as well as across datasets.

5.3.5 *Occlusion estimation*

We finally evaluate the accuracy of occlusion estimation on the Sintel training set as no public benchmarks are available for the task. Table 5.7 shows the comparison with state-of-the-art algorithms. Supervised methods are trained on FlyingChairs

Method	Training		Test
	AEPE	Fl-all	Fl-All
Separable flow† (Zhang et al., 2021)	(0.69)	(1.60%)	**4.64%**
RAFT-A† (Sun et al., 2021)	–	–	4.78%
RAFT† (Teed and Deng, 2020)	(0.63)	(1.5%)	5.10%
GMA† (Jiang et al., 2021)	(0.57)	(1.2%)	5.15%
MaskFlowNet (Zhao et al., 2020)	–	–	6.11%
VCN (Yang and Ramanan, 2019)	(1.16)	(4.1%)	6.30%
MFF§ (Ren et al., 2019)	–	–	7.17%
IRR-PWC (Ours)	(1.63)	(5.32%)	7.65%
PWC-Net+ (Sun et al., 2020)	(1.45)	(7.59%)	7.72%
LiteFlowNet (Hui et al., 2018)	(1.62)	(5.58%)	9.38%
PWC-Net (Sun et al., 2018)	(2.16)	(9.80%)	9.60%
ContinualFlow_ROB†§ (Neoral et al., 2018)	–	–	10.03%
MirrorFlow (Hur and Roth, 2017)	–	9.98%	10.29%
FlowNet2 (Ilg et al., 2017)	(2.30)	(8.61%)	10.41%
SDF Bai et al., 2016	–	12.14%	11.01%
SfM-PM§ Maurer et al., 2018	4.16	13.61%	11.83%
MR-Flow§ Wulff et al., 2017	–	14.09%	12.19%

Table 5.6: KITTI **Optical Flow 2015**: Average End-Point Error (AEPE) and outlier rates (Fl-Noc and Fl-all).

and FlyingThings3D; unsupervised methods are trained on Sintel without the use of ground truth. We achieve state-of-the-art accuracy with far fewer parameters (6.00M instead of 110M) and much simpler training schedules than the previous state of the art (Ilg et al., 2018).

5.3.6 *Qualitative Comparison*

Occlusion estimation. Figure 5.12 demonstrates a qualitative comparison with the state of the art on occlusion estimation. Qualitatively, MirrorFlow (Hur and Roth, 2017) misses many occlusions in general, and FlowNet-CSSR-ft-sd (Ilg et al., 2018) is able to detect fine details of occlusion. In contrast, our method tries not to miss occlusions, which results in a better recall rate but somewhat lower precision than those of FlowNet-CSSR-ft-sd (Ilg et al., 2018). Overall, our method demonstrates better F1-score than FlowNet-CSSR-ft-sd (Ilg et al., 2018), achieving state-of-the-art results on the evaluation dataset (*i. e.*, Sintel Train Clean and Final). Note that FlowNet-CSSR-ft-sd (Ilg et al., 2018) is additionally trained on the ChairsSDHom dataset (Ilg et al., 2017) for handling small-displacement motion, which is related to thinly-shaped occlusions. Our approach is not trained further.

Bi-directional flows and occlusion maps. MirrorFlow (Hur and Roth, 2017) is one of the most recent related works that estimates bi-directional flow and occlusion maps. Fig. 5.13 provides a qualitative comparison with MirrorFlow (Hur and Roth, 2017) on

Method	Type	Sintel Training	
		Clean	Final
IRR-PWC (Ours)	supervised	**0.712**	**0.669**
FlowNet-CSSR (Ilg et al., 2018)	supervised	0.703	0.654
OccAwareFlow (Wang et al., 2018)	unsupervised	0.54	0.48
Back2FutureFlow (Janai et al., 2018)	unsupervised	0.49	0.44
MirrorFlow (Hur and Roth, 2017)	estimated	0.390	–

Table 5.7: Occlusion estimation results on Sintel Training

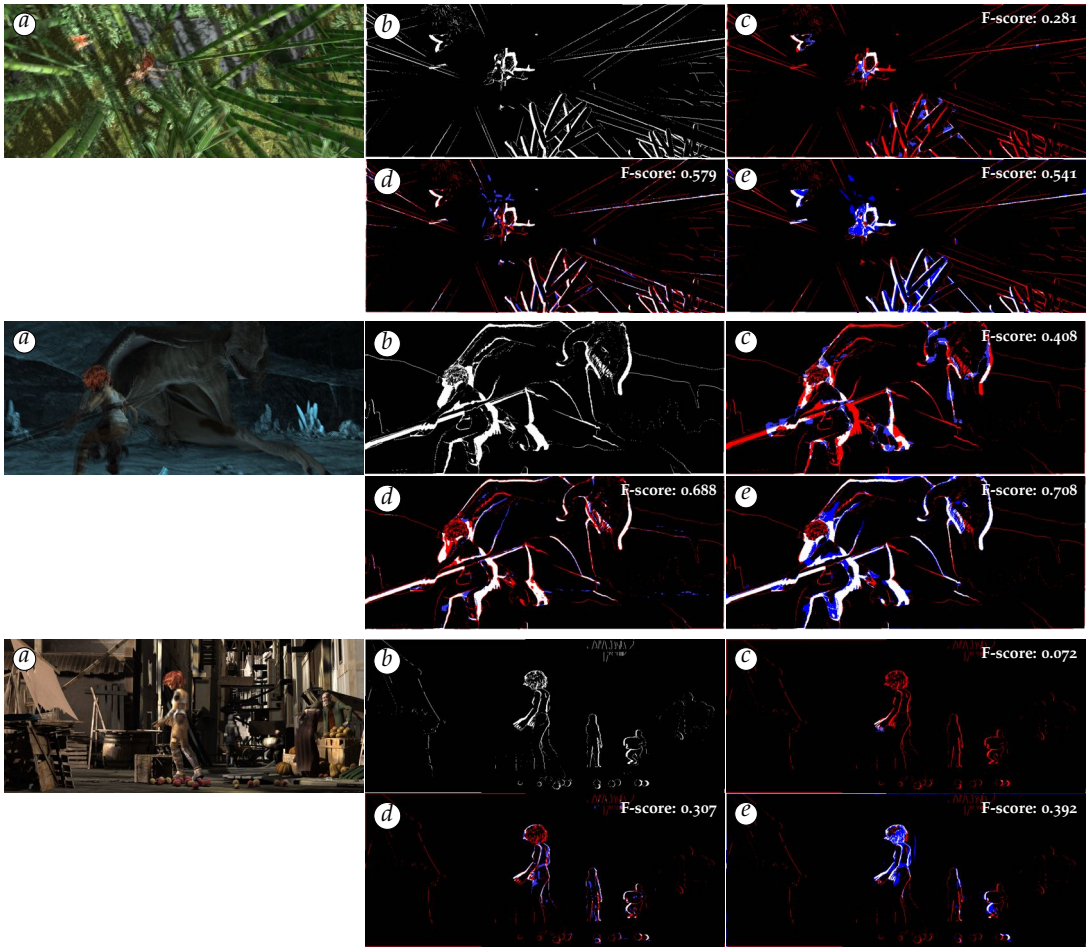


Figure 5.12: **Qualitative comparison of occlusion estimation with the state of the art**: *(a)* overlapped input images, *(b)* occlusion ground truth, *(c)* MirrorFlow (Hur and Roth, 2017), *(d)* FlowNet-CSSR-ft-sd (Ilg et al., 2018), and *(e)* ours. In the result image of each method, blue pixels denote **false positives**, red pixels denote **false negatives**, and white ones denote true positives (*i. e.*, correctly estimated occlusion). We include the F-score of each method in the top-right corner. Our model yields a better F-score on the second and the third scene than FlowNet-CSSR-ft-sd (Ilg et al., 2018).

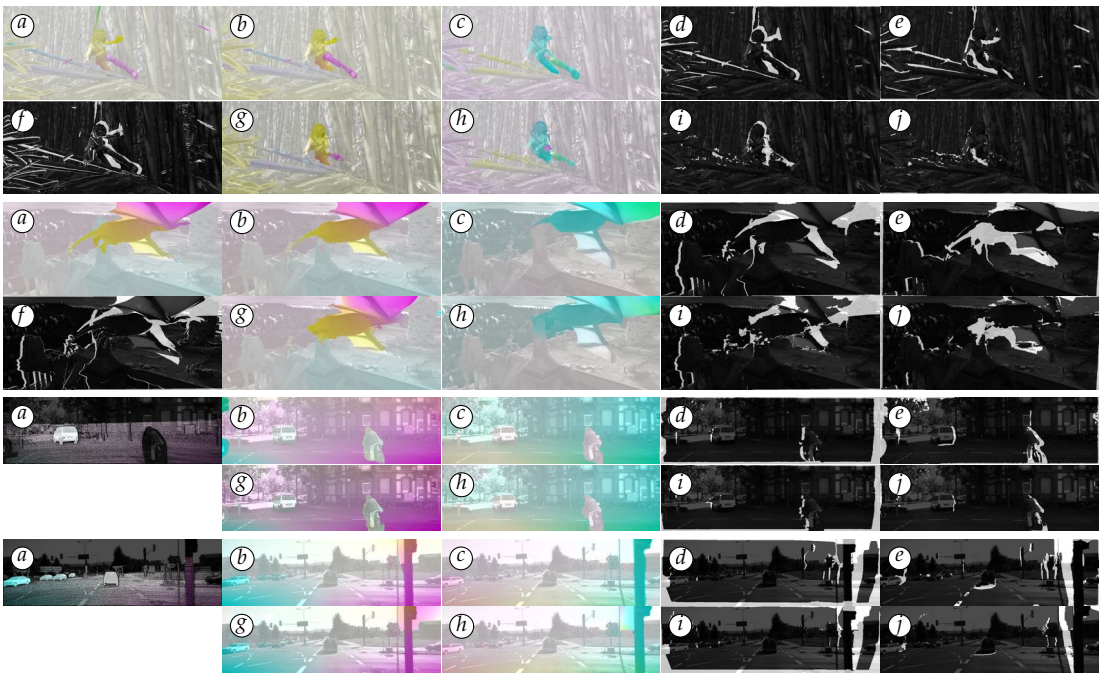


Figure 5.13: **Qualitative comparison of the bi-drectional optical flows and occlusion maps in both views with MirrorFlow (Hur and Roth, 2017)**: All results are overlayed on the corresponding image, either the first frame or the second frame. *(a)* Ground truth optical flow, *(b)* our forward flow, *(c)* our backward flow, *(d)* our occlusion map for the first frame, *(e)* our occlusion map for the second frame, *(f)* ground truth occlusion map, *(g)* forward flow of MirrorFlow, *(h)* backward flow of MirrorFlow, *(i)* occlusion map of MirrorFlow for the first frame, *(j)* occlusion map of MirrorFlow for the second frame. Note that KITTI has only sparse ground truth for optical flow and does not provide ground truth for occlusion.

the Sintel and KITTI 2015 datasets. In this comparison, we use our model fine-tuned on the training set of each dataset and display qualitative examples from the validation split. Comparing to MirrorFlow (Hur and Roth, 2017), our model demonstrates far fewer artifacts and fewer missing details for both flow and occlusion estimation. Although there is no ground truth for backward flow nor an occlusion map for the second image available for supervision, our bi-directional model is able to estimate the backward flow and the second occlusion map well while only using the ground truth of forward flow and the occlusion map for the first image (latter is only available on Sintel) during fine-tuning.

5.3.7 *Runtime analysis*

We provide how much our proposed schemes affect runtime on each backbone architecture. The runtime is measured on a single Tesla T4 GPU. Table 5.8 provides runtime of each major model on FlowNet backbone architecture. Our IRR version of FlowNet shows marginally faster runtime than the original FlowNet due to the change in the encoder part. Along the increase on the number of iteration steps, the

Model	Runtime (ms)
FlowNetS baseline	22
FlowNetS with IRR (1 iteration)	19
FlowNetS with IRR (2 iterations)	35
FlowNetS with IRR (3 iterations)	51
IRR-FlowNet (2 iterations)	264

Table 5.8: **Runtime analysis on FlowNet**: our IRR version of FlowNet is marginally faster than the original FlowNet, and the runtime linearly increases with the number of iteration steps.

Model	Runtime (ms)
PWC-Net baseline	62
PWC-Net with IRR	62
IRR-PWC	367

Table 5.9: **Runtime analysis on PWC-Net**: our IRR version of PWC-Net keeps the same runtime as the PWC-Net baseline. The runtime increase on our final model, IRR-PWC, comes the bi-directional estimation, the occlusion estimation, the bilateral refinement, and the occlusion upsampling layer.

runtime also linearly increases, by approximately 16 *(ms)* per each additional iteration step.

Table 5.9 provides the analysis on PWC-Net backbone architecture. Our IRR version of PWC-Net maintains the same runtime as the PWC-Net baseline because the number of iteration steps is the same as the number of pyramid levels. The runtime increase on our final model, IRR-PWC, comes the bi-directional estimation, the occlusion estimation, the bilateral refinement, and the occlusion upsampling layer.

5.4 DISCUSSION

In this chapter, we proposed an Iterative Residual Refinement (IRR) scheme based on weight sharing for generic optical flow networks, with additional components for bi-directional estimation and joint occlusion estimation. Application of our scheme on top of the two representative flow networks, FlowNet (Dosovitskiy et al., 2015) and PWC-Net (Sun et al., 2018), significantly improves flow accuracy with a better generalization, notably with the fewer number of parameters in case of PWC-Net (Sun et al., 2018). Especially as our second joint objective, the joint estimation of occlusion and optical flow results in accuracy improvements on both domains and set the state of the art on public benchmark datasets at the time of publication. We believe that our powerful IRR scheme can be combined with other baseline networks and provide the evidence base of other follow-up approaches, including multi-frame methods.

6

SELF-SUPERVISED MONOCULAR SCENE FLOW ESTIMATION

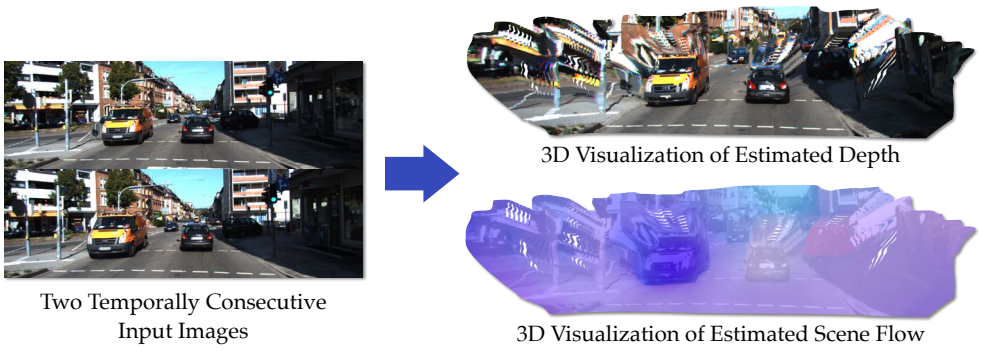


Figure 6.1: **Results of our monocular scene flow approach on the KITTI dataset (Geiger et al., 2013).** Given two consecutive images *(left)*, our method jointly predicts depth and scene flow *(right)*. (x,z)-coordinates of 3D scene flow are visualized using an optical flow color coding.

CONTENTS

As our last joint objective, we propose a scene flow approach that jointly estimates depth and 3D motion from two temporally consecutive monocular images. This monocular scene flow estimation, however, is a highly ill-posed problem, and practical solutions are lacking to date. In this chapter, we present a novel monocular scene flow method that yields competitive accuracy *and* real-time performance. Taking an inverse problem view, we design a single Convolutional Neural Network (CNN) that successfully estimates depth and 3D motion simultaneously from a classical optical flow cost volume. We adopt self-supervised learning with 3D loss functions and occlusion reasoning to leverage unlabeled data. We validate our design choices, including the proxy loss and augmentation setup. Our model achieves state-of-the-art accuracy among unsupervised/self-supervised learning approaches to monocular scene flow, and yields competitive results for the optical flow and monocular depth estimation sub-tasks. Semi-supervised fine-tuning further improves the accuracy and yields promising results in real-time.

6.1 INTRODUCTION

Scene flow estimation is the task of obtaining 3D structure and 3D motion of dynamic scenes, which is crucial to environment perception, *e. g.*, in the context of autonomous navigation. Consequently, many scene flow approaches have been proposed recently, based on different types of input data, such as stereo images (Huguet and Devernay, 2007; Schuster et al., 2018b; Vogel et al., 2013b; Wedel et al., 2011; Zhang and Kambhamettu, 2001), 3D point clouds (Gu et al., 2019; Liu et al., 2019d), or a sequence of RGB-D images (Hadfield and Bowden, 2011; Hornáček et al., 2014; Lv et al., 2018; Qiao et al., 2018; Quiroga et al., 2014; Thakur and Mukherjee, 2018).[1] However, each sensor configuration has its own limitations, *e. g.* stereo calibration required for a stereo rig, expensive sensing devices (*e. g.*, LiDAR) for 3D point measurement, or exclusive indoor usage (*i. e.*, RGB-D camera). We here consider *monocular 3D scene flow estimation*, aiming to overcome these limitations.

Monocular scene flow estimation, however, is a highly ill-posed problem, because both monocular depth (also called single-view depth) and per-pixel 3D motion need to be estimated from two temporally consecutive monocular frames. Relatively, few approaches have been suggested so far (Brickwedde et al., 2019; Xiao et al., 2017), and none of which achieves both reasonable accuracy and real-time performance.

Recently, a number of CNN approaches (Chen et al., 2019; Liu et al., 2019a; Luo et al., 2019; Ranjan et al., 2019; Yang et al., 2018; Zou et al., 2018b) have been proposed to jointly estimate depth, flow, and camera ego-motion in a monocular setup. This makes it possible to recover 3D motion from the various outputs, but has critical limitations, such as the depth–scale ambiguity (Ranjan et al., 2019; Zou et al., 2018b) and impossible depth estimation in occluded regions (Chen et al., 2019; Liu et al., 2019a; Luo et al., 2019; Yang et al., 2018), which significantly limit the ability to obtain accurate 3D scene flow across the entire image.

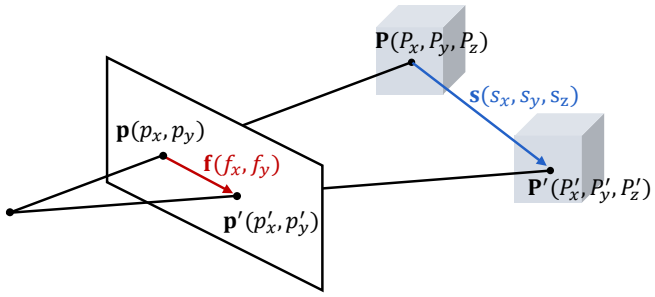
1 Please refer to Section 2.2 for a more comprehensive review on those methods.

In this chapter, we propose a CNNs-based monocular scene flow approach that yields competitive accuracy *and* real-time performance. To the best of our knowledge, our method is the first monocular scene flow method that directly predicts 3D scene flow from a CNN. Due to the scarcity of 3D motion ground truth and the domain over-fitting problem when using synthetic datasets (Butler et al., 2012; Mayer et al., 2016), we train directly on the target domain in a self-supervised manner to leverage large amounts of unlabeled data. Optional semi-supervised fine-tuning on limited quantities of ground-truth data can further boost the accuracy. In order to estimate depth and scene flow on an absolute scale, we utilize pairs of stereo images with their known configurations at training time. At test time, our method is purely monocular and needs only monocular images with known intrinsics.

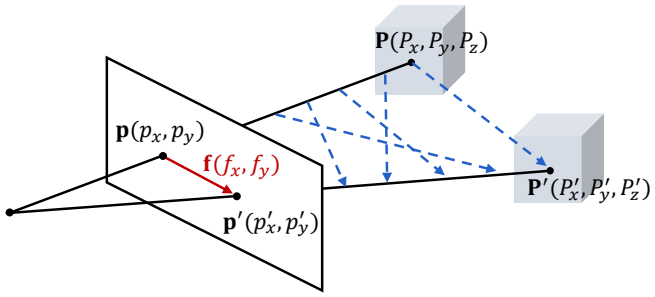
We make three main technical contributions: *(i)* We propose to approach this ill-posed problem by taking an *inverse problem view*. Noting that optical flow is the 2D projection of a 3D point and its 3D scene flow, we take the inverse direction and estimate scene flow in the monocular setting by *decomposing a classical optical flow cost volume into scene flow and depth* using a *single joint decoder*. We use a standard optical flow pipeline (PWC-Net (Sun et al., 2018)) as basis and adapt it for monocular scene flow. We verify our architectural choice and motivation by comparing with multi-task CNN approaches. *(ii)* We demonstrate that solving the monocular scene flow task with a single joint decoder actually *simplifies* joint depth and flow estimation methods (Chen et al., 2019; Liu et al., 2019a; Luo et al., 2019; Ranjan et al., 2019; Yang et al., 2018; Zou et al., 2018b), and yields competitive accuracy despite a simpler network. Existing multi-task CNN methods have multiple modules for the various tasks and often require complex training schedules due to the instability of training multiple CNNs jointly. In contrast, our method only uses a single network that outputs scene flow and depth (as well as optical flow after projecting to 2D) with a *simpler training setup* and better accuracy for depth and scene flow. *(iii)* We introduce a *self-supervised loss function* for monocular scene flow as well as a suitable *data augmentation scheme*. We introduce a view synthesis loss, a 3D reconstruction loss, and an occlusion-aware loss, all validated in an ablation study. Interestingly, we find that the geometric augmentations of the two tasks conflict with one another and determine a suitable compromise using an ablation study.

After training on unlabeled data from the KITTI raw dataset (Geiger et al., 2013), we evaluate on the KITTI Scene Flow dataset (Menze et al., 2015b; 2018) and demonstrate highly competitive accuracy compared to previous unsupervised/self-supervised learning approaches to monocular scene flow (Luo et al., 2019; Yang et al., 2018; Yin and Shi, 2018), increasing the accuracy by 34.0%. The accuracy of our fine-tuned network moves even closer to that of the semi-supervised method of Brickwedde et al. (2019), while being orders of magnitude faster.

6.2 SELF-SUPERVISED MONOCULAR SCENE FLOW



(a) Projecting scene flow into 2D space.



(b) Back-projecting optical flow into 3D space.

Figure 6.2: **Relating monocular scene flow estimation to optical flow**: *(a)* Projection of scene flow into the image plane yields optical flow (Yan and Xiang, 2016). *(b)* Back-projection of optical flow leaves an ambiguity in jointly determining depth and scene flow.

6.2.1 *Problem formulation*

For each pixel $\mathbf{p} = (p_x, p_y)$ in the reference frame $\mathbf{I}_t$, our main objective is to estimate the corresponding 3D point $\mathbf{P} = (P_x, P_y, P_z)$ and its (forward) scene flow $\mathbf{s} = (s_x, s_y, s_z)$ to the target frame $\mathbf{I}_{t+1}$, as illustrated in Fig. 6.2a. The scene flow is defined as 3D motion with respect to the camera, and its projection onto the image plane becomes the optical flow $\mathbf{f} = (f_x, f_y)$.

To estimate scene flow in the monocular camera setting, we take an inverse problem approach: we use CNNs to estimate a classical *optical flow cost volume* as an intermediate representation, which is then *decomposed* with a *learned decoder* into 3D points and their scene flow. Unlike scene flow with a stereo camera setup (Lai et al., 2019; Lee et al., 2019; Wang et al., 2019b), it is challenging to determine depth on an absolute scale due to the scale ambiguity. Yet, relating per-pixel correspondence between two images can provide a cue for estimating depth in the monocular setting. Also, given an optical flow estimate, back-projecting optical flow into 3D yields many possible combinations of depth and scene flow, see Fig. 6.2b, which makes the problem much more challenging.

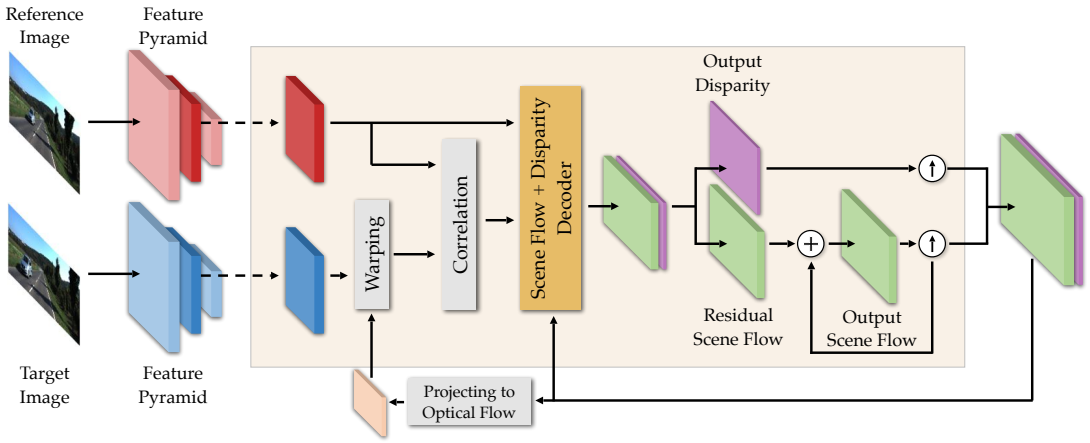


Figure 6.3: **Our monocular scene flow architecture based on PWC-Net (Sun et al., 2018)**: while maintaining the overall original structure of PWC-Net, we modify the decoder to output *residual scene flow* and (non-residual) *disparity* together. After the residual update of scene flow, we project the scene flow back to optical flow using depth. Then, the optical flow is used for warping the feature map (only 3 of 7 levels shown for ease of visualization) in the next pyramid level. The light-yellow shaded region shows one forward pass for each pyramid level.

6.2.2 *Network architecture*

In contrast to previous work (Chen et al., 2019; Luo et al., 2019; Ranjan et al., 2019; Yang et al., 2018; Yin and Shi, 2018; Zou et al., 2018b) that uses separate networks for each task (*e. g.*, optical flow, depth, and camera motion), our method only uses one single CNN model that outputs both 3D scene flow and disparity[2] through a *single decoder*. We argue that having a single decoder is more sensible in our monocular setting than separate decoders, because when decomposing evidence for 2D correspondence into 3D structure and 3D motion, their interplay needs to be taken into account (*cf*. Fig. 6.2b).

The first technical basis of our CNN model is PWC-Net (Sun et al., 2018), one of the state-of-the-art optical flow networks, which we modify for our task. Fig. 6.3 illustrates our monocular scene flow architecture atop PWC-Net. PWC-Net has a pyramidal structure that constructs a feature pyramid and incrementally updates the estimation across the pyramid levels. The yellow-shaded area shows one forward pass for each pyramid level.

While maintaining the original structure, we increase the number of feature channels in the pyramidal feature extractor from $[16, 32, 64, 96, 128, 196]$ to $[32, 64, 96, 128, 192, 256]$ in order to have more discriminate features. We modify the decoder of each pyramid level to output disparity and scene flow together by increasing the number of output channels from 2 to 4 (*i. e.*, 3 for scene flow and 1 for disparity). Following the benefit of residual motion estimation in the context of optical flow (Hui et al., 2018; Hur and

2 Even though we do not have stereo images at test time, we still estimate disparity of a hypothetical stereo setup following Godard et al. (2019, 2017), which can be converted into depth given the assumed stereo configuration.

Roth, 2019; Sun et al., 2018), we estimate residual scene flow at each level. In contrast, we observe that residual updates hurt disparity estimation, hence we estimate (non-residual) disparity at all levels. We conjecture that, unlike optical flow or scene flow, learning to residually update disparity from a previous pyramid level is a harder task than learning to output just non-residual disparity at each level. This conclusion may depend on architectural chocies and warrants further investigation.

6.2.3 *Addressing the scale ambiguity*

When resolving the 3D ambiguities, it is not possible to determine the depth scale from a single correspondence in two monocular images. In order to estimate depth and scene flow on an *absolute scale*, we adopt the monocular depth estimation approach of Godard et al. (2019, 2017) as our second basis, which utilizes pairs of stereo images with their known stereo configuration and camera intrinsics $\mathbf{K}$ for training; at test time, only monocular images and known intrinsics are needed. The core idea is to let a network remember an absolute scale information of the training dataset, while assuming that the same camera configuration is used at test time. The input of the network is images from the left camera only. During training time, corresponding images from the right camera guide the CNN to estimate the hypothetical disparity d of the left images on an absolute scale by exploiting semantic and geometric cues indirectly (Dijk and Croon, 2019) through a self-supervised loss function. Then the depth $\hat{d}$ can be trivially recovered given the baseline distance of a stereo rig b and the camera focal length f_{focal} as $\hat{d} = b \cdot f_{\text{focal}} / d$. We also use stereo images only for training; at test time our approach is *purely monocular*. In our context, estimating depth on an absolute scale helps to disambiguate scene flow on an absolute scale as well (*cf*. Fig. 6.2b). Moreover, tightly coupling temporal correspondence and depth actually helps to identify the appropriate absolute scale, which allows us to avoid unrealistic testing settings that other monocular methods rely on (*e. g.*, Ranjan et al. (2019), Yin and Shi (2018), and Zou et al. (2018b) use *ground truth* to correctly scale their predictions at test time). However, one major limitation of memorizing the absolute scale of the training dataset is about generalization; correct scale estimation in other datasets with different camera configurations is not possible, and the estimation will be only up to scale. We leave scale-aware estimation on multiple datasets as future work.

6.2.4 *A proxy loss for self-supervised learning*

Similar to previous monocular structure reconstruction methods (Chen et al., 2019; Luo et al., 2019; Ranjan et al., 2019; Yang et al., 2018; Yin and Shi, 2018; Zhu et al., 2019; Zou et al., 2018b), we exploit a view synthesis loss to guide the network to jointly estimate disparity and scene flow. For better accuracy in both tasks, we exploit occlusion cues through bi-directional estimation (Meister et al., 2018), here of disparity and scene flow. Given a stereo image pair of the reference and target frame $\{\mathbf{I}^{\text{l}}_t, \mathbf{I}^{\text{l}}_{t+1}, \mathbf{I}^{\text{r}}_t, \mathbf{I}^{\text{r}}_{t+1}\}$, we input a monocular sequence from the left camera ($\mathbf{I}^{\text{l}}_t$ and $\mathbf{I}^{\text{l}}_{t+1}$) to the

Figure 6.4: **Illustration of disparity photometric loss**: The disparity photometric loss penalizes the photometric difference between the left image and the reconstructed left image only for visible pixels.

network and obtain a disparity map of each frame (d_t^{l} and d_{t+1}^{l}) as well as forward and backward scene flow ($\mathbf{s}_{\mathrm{fw}}^{\mathrm{l}}$ and $\mathbf{s}_{\mathrm{bw}}^{\mathrm{l}}$) by simply switching the temporal order of the input. The two images from the right camera ($\mathbf{I}_t^{\mathrm{r}}$ and $\mathbf{I}_{t+1}^{\mathrm{r}}$) are used only as a guidance in the loss function and are not used at test time. Our total loss is a weighted sum of a disparity loss L_{d} and a scene flow loss L_{sf},

$$L_{\mathrm{total}} = L_{\mathrm{d}} + \lambda_{\mathrm{sf}} L_{\mathrm{sf}}. \tag{6.1}$$

Disparity loss. Based on the approach of Godard Godard et al. (2019, 2017), we propose an occlusion-aware monocular disparity loss, consisting of a photometric loss $L_{\mathrm{d\_ph}}$ and a smoothness loss $L_{\mathrm{d\_sm}}$,

$$L_{\mathrm{d}} = L_{\mathrm{d\_ph}} + \lambda_{\mathrm{d\_sm}} L_{\mathrm{d\_sm}}, \tag{6.2}$$

with regularization parameter $\lambda_{\mathrm{d\_sm}} = 0.1$. The disparity loss is applied to both disparity maps d_t^{l} and d_{t+1}^{l}. For brevity, we only describe the case of d_t^{l}.

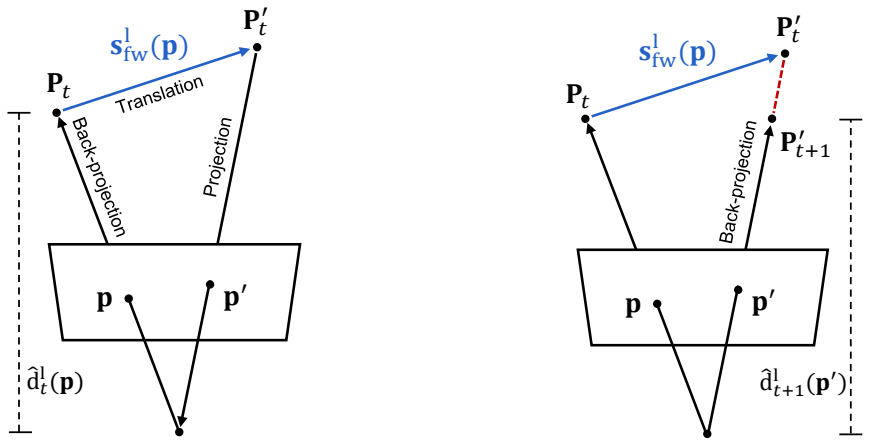
The *photometric loss* $L_{\mathrm{d\_ph}}$ penalizes the photometric difference between the left image $\mathbf{I}_t^{\mathrm{l}}$ and the reconstructed left image $\tilde{\mathbf{I}}_t^{\mathrm{l,d}}$, which is synthesized from the output disparity map d_t^{l} and the given right image $\mathbf{I}_t^{\mathrm{r}}$ using bilinear interpolation (Jaderberg et al., 2015). Different to Godard et al. (2019, 2017), we only penalize the photometric loss for non-occluded pixels as illustrated in Fig. 6.4. Following standard practice (Godard et al., 2019; 2017), we use a weighted combination of an L_1 loss and the structural similarity index (SSIM) (Wang et al., 2004):

$$L_{\mathrm{d\_ph}} = \frac{\sum_{\mathbf{p}} \left(1 - O_t^{\mathrm{l,disp}}(\mathbf{p})\right) \cdot \rho\left(\mathbf{I}_t^{\mathrm{l}}(\mathbf{p}), \tilde{\mathbf{I}}_t^{\mathrm{l,d}}(\mathbf{p})\right)}{\sum_{\mathbf{q}} \left(1 - O_t^{\mathrm{l,disp}}(\mathbf{q})\right)} \tag{6.3a}$$

with

$$\rho(a,b) = \alpha \frac{1 - \mathrm{SSIM}(a,b)}{2} + (1-\alpha)\|a-b\|_1, \tag{6.3b}$$

where $\alpha = 0.85$ and $O_t^{\mathrm{l,disp}}$ is the disparity occlusion mask (0 – visible, 1 – occluded). To obtain the occlusion mask $O_t^{\mathrm{l,disp}}$, we feed the right image $\mathbf{I}_t^{\mathrm{r}}$ into the network to obtain the right disparity d_t^{r} and take the inverse of its disocclusion map, which is obtained by forward-warping the right disparity map (Hur and Roth, 2017; Wang et al., 2018).



(a) Photometric loss. (b) 3D point reconstruction loss.

Figure 6.5: **Scene flow losses**: *(a)* Finding corresponding pixels given depth and scene flow for the photometric loss $L_{\mathrm{sf\_ph}}$ (Eq. (6.7)). *(b)* Penalizing 3D distance (dashed, red) between corresponding 3D points by the point reconstruction loss $L_{\mathrm{sf\_pt}}$ (Eq. (6.8)).

To encourage locally smooth disparity estimates, we adopt an *edge-aware* 2^{nd}*-order smoothness* (Liu et al., 2019a; Meister et al., 2018; Woodford et al., 2008),

$$L_{\mathrm{d\_sm}} = \frac{1}{N} \sum_{\mathbf{p}} \sum_{i \in \{x,y\}} \left| \nabla_i^2 d_t^{\mathrm{l}}(\mathbf{p}) \right| \cdot e^{-\beta \| \nabla_i \mathbf{I}_t^{\mathrm{l}}(\mathbf{p}) \|_1}, \tag{6.4}$$

with $\beta = 10$ and N being the number of pixels.

Scene flow loss. The scene flow loss consists of three terms – a photometric loss $L_{\mathrm{sf\_ph}}$, a 3D point reconstruction loss $L_{\mathrm{sf\_pt}}$, and a scene flow smoothness loss $L_{\mathrm{sf\_sm}}$,

$$L_{\mathrm{sf}} = L_{\mathrm{sf\_ph}} + \lambda_{\mathrm{sf\_pt}} L_{\mathrm{sf\_pt}} + \lambda_{\mathrm{sf\_sm}} L_{\mathrm{sf\_sm}}, \tag{6.5}$$

with regularization parameters $\lambda_{\mathrm{sf\_pt}} = 0.2$ and $\lambda_{\mathrm{sf\_sm}} = 200$.[3] The scene flow loss is applied to both forward and backward scene flow ($\mathbf{s}^{\mathrm{l}}_{\mathrm{fw}}$ and $\mathbf{s}^{\mathrm{l}}_{\mathrm{bw}}$). Again for brevity, we only describe the case of forward scene flow $\mathbf{s}^{\mathrm{l}}_{\mathrm{fw}}$.

The *scene flow photometric loss* $L_{\mathrm{sf\_ph}}$ penalizes the photometric difference between the reference image $\mathbf{I}^{\mathrm{l}}_t$ and the reconstructed reference image $\tilde{\mathbf{I}}^{\mathrm{l,sf}}_t$, synthesized from the disparity map d^{l}_t, the output scene flow $\mathbf{s}^{\mathrm{l}}_{\mathrm{fw}}$, and the target image $\mathbf{I}^{\mathrm{l}}_{t+1}$ (*cf*. Fig. 6.6a). To reconstruct the image, the corresponding pixel coordinate $\mathbf{p}'$ in $\mathbf{I}^{\mathrm{l}}_{t+1}$ of each pixel $\mathbf{p}$ in $\mathbf{I}^{\mathrm{l}}_t$ is calculated by back-projecting the pixel $\mathbf{p}$ into 3D space using the camera intrinsics $\mathbf{K}$ and estimated depth $\hat{d}^{\mathrm{l}}_t(\mathbf{p})$, translating the points using the scene flow $\mathbf{s}^{\mathrm{l}}_{\mathrm{fw}}(\mathbf{p})$, and then re-projecting them to the image plane (*cf*. Fig. 6.5a),

$$\mathbf{p}' = \mathbf{K} \Big(\hat{d}^{\mathrm{l}}_t(\mathbf{p}) \cdot \mathbf{K}^{-1} \mathbf{p} + \mathbf{s}^{\mathrm{l}}_{\mathrm{fw}}(\mathbf{p}) \Big), \tag{6.6}$$

3 We provide a study on the choice of hyper-parameter in Appendix C.3.

Reference image | Reconstructed reference image | Disocclusion map obtained from backward scene flow

(a) Photometric loss.

Target 3D points | Reconstructed target 3D points | Disocclusion map obtained from backward scene flow

(b) 3D point reconstruction loss.

Figure 6.6: **Illustration of scene flow loss**: *(a)* The scene flow photometric loss (Eq. (6.7)) penalizes the photometric difference between the reference image and the reconstructed reference image. *(b)* The 3D point reconstruction loss (Eq. (6.8)) penalizes the Euclidean distance between the two corresponding 3D points. Both losses are only applied on visible pixels.

assuming a homogeneous coordinate representation. Then, we apply the same occlusion-aware photometric loss as in the disparity case (Eq. (6.3a)),

$$L_{\text{sf\_ph}} = \frac{\sum_{\mathbf{p}} \left(1 - O_t^{\text{l,sf}}(\mathbf{p})\right) \cdot \rho\left(\mathbf{I}_t^{\text{l}}(\mathbf{p}), \tilde{\mathbf{I}}_t^{\text{l,sf}}(\mathbf{p})\right)}{\sum_{\mathbf{q}} \left(1 - O_t^{\text{l,sf}}(\mathbf{q})\right)}, \tag{6.7}$$

where $O_t^{\text{l,sf}}$ is the scene flow occlusion mask, obtained by calculating disocclusion using the backward scene flow $\mathbf{s}_{\text{bw}}^{\text{l}}$.

Additionally, as visualized in Fig. 6.6b, we also penalize the Euclidean distance between the two corresponding 3D points, *i. e.* the translated 3D point of pixel $\mathbf{p}$ from the reference frame and the matched 3D point in the target frame (*cf*. Fig. 6.5b):

$$L_{\text{sf\_pt}} = \frac{\sum_{\mathbf{p}} \left(1 - O_t^{\text{l,sf}}(\mathbf{p})\right) \cdot \left\| \mathbf{P}'_t - \mathbf{P}'_{t+1} \right\|_2}{\sum_{\mathbf{q}} \left(1 - O_t^{\text{l,sf}}(\mathbf{q})\right)}, \tag{6.8a}$$

with

$$\mathbf{P}'_t = \hat{d}_t^{\text{l}}(\mathbf{p}) \cdot \mathbf{K}^{-1}\mathbf{p} + \mathbf{s}_{\text{fw}}^{\text{l}}(\mathbf{p}) \tag{6.8b}$$

$$\mathbf{P}'_{t+1} = \hat{d}_{t+1}^{\text{l}}(\mathbf{p}') \cdot \mathbf{K}^{-1}\mathbf{p}', \tag{6.8c}$$

and $\mathbf{p}'$ as defined in Eq. (6.6). Again, this *3D point reconstruction loss* is only applied on visible pixels, where the correspondence should hold.

Analogous to the disparity loss in Eq. (6.4), we also adopt *edge-aware* 2^{nd}*-order smoothness for scene flow* to encourage locally smooth estimation:

$$L_{\text{sf\_sm}} = \frac{1}{N} \sum_{\mathbf{p}} \sum_{i \in \{x,y\}} \left| \nabla_i^2 \mathbf{s}_{\text{fw}}^{\text{l}}(\mathbf{p}) \right| \cdot e^{-\beta \|\nabla_i \mathbf{I}_t^{\text{l}}(\mathbf{p})\|_1}. \tag{6.9}$$

		Monocular depth		Monocular scene flow			
Aug.	CC.	Abs. Rel.	Sq. Rel.	D1-all	D2-all	Fl-all	SF-all
		0.113	1.118	32.06	36.46	24.68	49.89
✓		0.122	1.172	**31.25**	**34.86**	**23.49**	**47.05**
	✓	**0.112**	**1.089**	37.24	39.26	24.82	54.83
✓	✓	0.121	1.155	33.25	36.21	24.73	49.12

Table 6.1: **Impact of geometric augmentations (*Aug.*) and CAM-Convs (*CC.*) (Facil et al., 2019) on monocular depth and scene flow estimation** (on KITTI split, see text): the accuracy of monocular depth estimation improves only when using CAM-Convs while that of monocular scene flow estimation improves when only using augmentation without CAM-Convs.

6.2.5 *Data augmentation*

In many prediction tasks, data augmentation is crucial to achieving good accuracy given limited training data. In our monocular scene flow task, unfortunately, the typical geometric augmentation schemes of the two tasks (*i. e.*, monocular depth estimation, scene flow estimation) conflict each other. For monocular depth estimation, not performing geometric augmentation is desirable as it enables learning the scene layout under a fixed camera configuration (Dijk and Croon, 2019; Hu et al., 2019). On the other hand, the scene flow necessitates geometric augmentations to match corresponding pixels better (Jiang et al., 2019; Mayer et al., 2016).

We investigate which type of (geometric) augmentation is suitable for our monocular scene flow task and method. Similar to previous multi-task approaches (Chen et al., 2019; Ranjan et al., 2019; Zou et al., 2018b), we prepare a simple data augmentation scheme, consisting of random scales, cropping, resizing, and horizontal image flipping. Upon the augmentation, we also explore the recent CAM-Convs (Facil et al., 2019), which facilitate depth estimation irrespective of the camera intrinsics. After applying augmentations on the input images, we calculate the resulting camera intrinsics and then input them in the format of CAM-Convs (see Facil et al. (2019) for technical details). We conjecture that using geometric augmentation will improve the scene flow accuracy. Yet, at the same time adopting CAM-Convs (Facil et al., 2019) could prevent the depth accuracy from dropping due to the changes in camera intrinsics of the augmented images. We conduct our empirical study on the *KITTI split* (Godard et al., 2017) of the KITTI raw dataset (Geiger et al., 2013) (see Section 6.3.1 for details).

Empirical study for monocular depth estimation. We use a ResNet18-based monocular depth baseline (Godard et al., 2017) using our proposed occlusion-aware loss. Table 6.1 (left hand side) shows the results. As we can see, geometric augmentations deteriorate the depth accuracy, since they prevent the network from learning a specific camera prior by inputting augmented images with diverse camera intrinsics; this observation holds with and without CAM-Convs. This likely explains why some

multi-task approaches (Lai et al., 2019; Lee et al., 2019; Liu et al., 2019a; Wang et al., 2019b) only use minimal augmentation schemes such as image flipping and input temporal-order switching. Only using CAM-Convs (Facil et al., 2019) works best as the test dataset contains images with different intrinsics, which CAM-Convs can handle.

Empirical study for monocular scene flow estimation. We train our full model with the proposed loss from Eq. (6.1). Looking at the right side of Table 6.1 yields different conclusions for monocular scene flow estimation: *using augmentation improves the scene flow accuracy in general, but using CAM-Convs (Facil et al., 2019) actually hurts the accuracy*. We conjecture that the benefit of CAM-Convs – introducing a test-time dependence on input camera intrinsics – may be redundant for correspondence tasks (*i. e.* optical flow, scene flow) and can hurt the accuracy. We also observe that CAM-Convs lead to slight over-fitting on the training set, yielding marginally lower training loss (*e. g.*, $< 1\%$) but with higher error on the test set. Therefore, we apply only geometric augmentation without CAM-Convs in the following.

6.3 EXPERIMENTS

6.3.1 *Implementation details*

Dataset. For evaluation, we use the KITTI raw dataset (Geiger et al., 2013), which provides stereo sequences covering 61 street scenes. For the scene flow experiments, we use the *KITTI Split* (Godard et al., 2017): we first exclude 29 scenes contained in *KITTI Scene Flow Training* (Menze et al., 2015b; 2018) and split the remaining 32 scenes into 25 801 sequences for training and 1684 for validation. For evaluation and the ablation study, we use *KITTI Scene Flow Training* as test set, since it provides ground-truth labels for disparity and scene flow for 200 images.

After training on *KITTI Split* in a self-supervised manner, we optionally fine-tune our model using *KITTI Scene Flow Training* (Menze et al., 2015b; 2018) to see how much accuracy gain can be obtained from annotated data. We fine-tune our model in a semi-supervised setting by combining a supervised loss with our self-supervised loss (see below for details).

Additionally for evaluating monocular depth accuracy, we also use the *Eigen Split* (Eigen et al., 2014) by excluding 28 scenes that the 697 test sequences cover, splitting into 20 120 training sequences and 1338 validation sequences.

Data augmentation. We adopt photometric augmentations with random gamma, brightness, and color changes. As discussed in Section 6.2.5, we use geometric augmentations consisting of horizontal flips (Lai et al., 2019; Lee et al., 2019; Liu et al., 2019a; Wang et al., 2019b), random scales, random cropping (Chen et al., 2019; Ranjan et al., 2019; Zou et al., 2018b), and then resizing into 256×832 pixels as in previous work (Lee et al., 2019; Liu et al., 2019a; Luo et al., 2019; Ranjan et al., 2019; Yang et al., 2018). Please refer to Appendix C.2 for details on the augmentation settings.

Self-supervised training. Our network is trained using Adam (Kingma and Ba, 2015) with hyper-parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Our initial learning rate is 2×10^{-4}, and the mini-batch size is 4. We train our network for a total of 400k iterations.[4] In every iteration, the regularization weight λ_{sf} in Eq. (6.1) is dynamically determined to make the loss of the scene flow and disparity be equal in order to balance the optimization of the two joint tasks (Hur and Roth, 2019). Our specific learning rate schedule, as well as details on hyper-parameter choice and data augmentation are provided in Appendices C.1 and C.2.

Unlike previous approaches requiring stage-wise pre-training (Lee et al., 2019; Liu et al., 2019a; Wang et al., 2019b; Zou et al., 2018b) or iterative training (Luo et al., 2019; Ranjan et al., 2019; Yang et al., 2018) of multiple CNNs due to the instability of joint training, our approach does not need any complex training strategies, but can just be trained from scratch all at once. *This highlights the practicality of our method.*

Semi-supervised fine-tuning. We optionally fine-tune our trained model in a semi-supervised manner by mixing the two datasets, the KITTI raw dataset (Geiger et al., 2013) and *KITTI Scene Flow Training* (Menze et al., 2015b; 2018), at a ratio of 3 : 1 in each batch of 4. The latter dataset provides sparse ground truth of the disparity map of the reference image, disparity information at the target image mapped into the reference image, as well as optical flow. We apply our self-supervised loss to all samples and a supervised loss (L_2 for optical flow, L_1 for disparity) only for the sample from *KITTI Scene Flow Training* after converting the scene flow into two disparity maps and optical flow. Through semi-supervised fine-tuning, the proxy loss can guide pixels that the sparse ground truth cannot supervise. Moreover, the model can be prevented from heavy over-fitting on the only 200 annotated images by leveraging more data. We train the network for 45k iterations with the learning rate starting at 4×10^{-5} (see Appendices C.1 and C.2).

Evaluation metric. For evaluating the scene flow accuracy, we follow the evaluation metric of *KITTI Scene Flow benchmark* (Menze et al., 2015b; 2018). It evaluates the accuracy of the disparity for the reference frame *(D1-all)* and for the target image mapped into the reference frame *(D2-all)*, as well as of the optical flow *(Fl-all)*. Each pixel that exceeds a threshold of 3 pixels and 5% *w. r. t.* the ground-truth disparity or optical flow is regarded as an outlier; the metric reports the outlier ratio (in %) among all pixels with available ground truth. Furthermore, if a pixel satisfies all metrics (*i. e.*, D1-all, D2-all, and Fl-all), it is regarded as valid scene flow estimate from which the outlier rate for scene flow *(SF-all)* is calculated. For evaluating the depth accuracy, we follow the standard evaluation scheme introduced by Eigen et al. (2014). We assume known test-time camera intrinsics.

6.3.2 *Ablation study*

To confirm the benefit of our various contributions, we conduct ablation studies based on our full model using the *KITTI split* with data augmentation applied.

4 Code is available at https://github.com/visinf/self-mono-sf.

Occ.	3D points	D1-all	D2-all	Fl-all	SF-all
	(Basic)	33.31	51.33	24.74	64.05
✓		**30.99**	50.89	23.55	62.50
	✓	32.07	36.01	27.30	49.27
✓	✓	31.25	**34.86**	**23.49**	**47.05**

Table 6.2: **Ablation study on the loss function**: based on the *Basic* 2D loss consisting of photometric and smoothness loss, the 3D point reconstruction loss (*3D points*) improves scene flow accuracy, especially when discarding occluded pixels in the loss (*Occ.*).

Proxy loss for self-supervised learning. Our proxy loss consists of three main components: *(i) Basic*: a basic combination of 2D photometric and smoothness losses, *(ii) 3D points*: the 3D point reconstruction loss for scene flow, and *(iii) Occ.*: whether applying the photometric and point reconstruction loss only for visible pixels or not. Table 6.2 shows the contribution of each loss toward the accuracy.

The *3D points* loss significantly contributes to more accurate scene flow by yielding more accurate disparity on the target image (D2-all). This highlights the importance of penalizing the actual 3D Euclidean distance between two corresponding 3D points (*cf*. Fig. 6.5b), which typical loss functions in 2D space (*i. e*. *Basic* loss) as in previous work (Luo et al., 2019; Yang et al., 2018) cannot.

Taking occlusion into account consistently improves the scene flow accuracy further. The main objective of our proxy loss is to reconstruct the reference image as closely as possible, which can lead to hallucinating potentially incorrect estimates of disparity and scene flow in the occluded areas. Thus, discarding occluded pixels in the loss is critical to achieving accurate predictions.

We provide a qualitative comparison of each setting in Appendix C.5.

Single decoder *vs*. separate decoders. To verify the key motivation of *decomposing optical flow cost volumes into depth and scene flow using a single decoder*, we compare against a model with separate decoders for each task, which follows the conventional design of other multi-task methods (Chen et al., 2019; Liu et al., 2019a; Luo et al., 2019; Ranjan et al., 2019; Yang et al., 2018; Zou et al., 2018b). We also prepare two baselines that estimate either monocular depth or optical flow only, to assess the capacity of our modified PWC-Net for each task.

Table 6.3 demonstrates our ablation study on the network design. First, our model with a single decoder achieves comparable or even higher accuracy on the depth and optical flow tasks, compared to using the same network only for each individual task. We thus conclude that solving monocular scene flow using a single joint network can substitute the two individual tasks given the same amount of training resources and network capacity.

When separating the decoders, we find that the network cannot be trained stably, yielding trivial solutions for disparity. This is akin to issues observed by previous multi-task approaches, which require pre-training or iterative training for multiple

Model	D1-all	D2-all	Fl-all	SF-all
Monocular depth only	**27.59**	–	–	–
Optical flow only	–	–	24.27	–
Scene flow w/ separate decoders	100	97.22	27.63	100
Scene flow w/ a single decoder	31.25	**34.86**	**23.49**	**47.05**

Table 6.3: **Single decoder *vs.* separate decoders**: using a single decoder yields stable training and comparable accuracy on both tasks to models that target each individual task separately.

Method	D1-all	D2-all	Fl-all	SF-all	Runtime
DF-Net (Zou et al., 2018b)	46.50	61.54	27.47	73.30	–
GeoNet (Yin and Shi, 2018)	49.54	58.17	37.83	71.32	0.06 s
EPC (Yang et al., 2018)	26.81	60.97	25.74	(>60.97)	0.05 s
EPC++ (Luo et al., 2019)	**23.84**	60.32	**19.64**	(>60.32)	0.05 s
Self-Mono-SF (Ours)	31.25	**34.86**	23.49	**47.05**	0.09 s
Mono-SF (Brickwedde et al., 2019)	16.72	18.97	11.85	21.60	41 s
Self-Mono-SF-ft (Ours)	(2.89)	(3.91)	(6.19)	(7.53)	0.09 s

Table 6.4: **Monocular scene flow evaluation on *KITTI Scene Flow Training***: our self-supervised learning approach significantly outperforms all multi-task CNN methods *(upper rows)* on the scene flow metric, *SF-all*. Lower rows provide the accuracy of a semi-supervised method (Brickwedde et al., 2019) and our fine-tuned model.

CNNs (Lee et al., 2019; Liu et al., 2019a; Luo et al., 2019; Ranjan et al., 2019; Wang et al., 2019b; Yang et al., 2018; Zou et al., 2018b). In contrast, having a single decoder resolves the imbalance and stability problem by virtue of joint estimation. We include a more comprehensive analysis in Appendix C.4, gradually splitting the decoder to closely analyze its behavior.

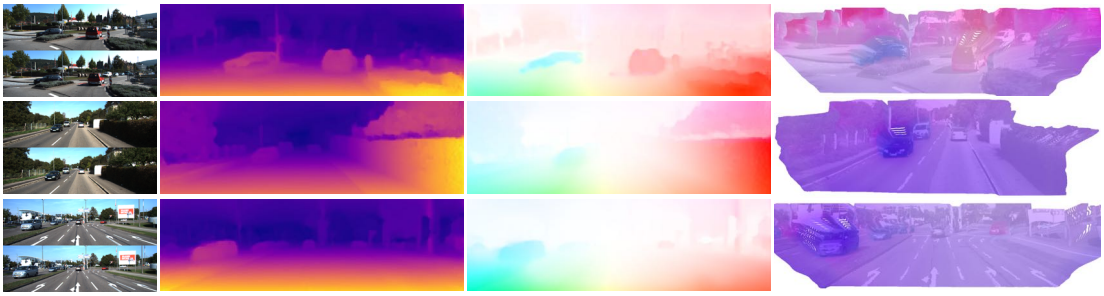
6.3.3 *Monocular scene flow*

Table 6.4 demonstrates the comparison to existing monocular scene flow methods on *KITTI Scene Flow Training*. We compare against state-of-the-art multi-task CNN methods (Luo et al., 2019; Yang et al., 2018; Yin and Shi, 2018; Zou et al., 2018b) on the scene flow evaluation metric. Our model significantly outperforms these methods by a large margin, confirming our method as the most accurate monocular scene flow method using CNNs to date. For example, our method yields more than 40.1% accuracy gain for estimating the disparity on the target image (D2-all). Though the two methods, EPC (Yang et al., 2018) and EPC++ (Luo et al., 2019), do not provide scene flow accuracy numbers (SF-all), we can conclude that our method clearly outperforms all four methods in SF-all, since SF-all is lower-bounded by D2-all.

Method	D1-all	D2-all	Fl-all	SF-all	Runtime
DRISF (Ma et al., 2019)	2.55	**4.04**	**4.73**	**6.31**	0.75 s
SENSE (Jiang et al., 2019)	**2.22**	5.89	7.64	9.55	0.32 s
PWOC-3D (Saxena et al., 2019)	5.13	8.46	12.96	15.69	0.13 s
UnOS (Wang et al., 2019b)	6.67	12.05	18.00	22.32	0.08 s
Mono-SF (Brickwedde et al., 2019)	**16.32**	**19.59**	**12.77**	**23.08**	41 s
Self-Mono-SF (Ours)	34.02	36.34	23.54	49.54	0.09 s
Self-Mono-SF-ft (Ours)	22.16	25.24	15.91	33.88	0.09 s

Table 6.5: **Scene flow evaluation on *KITTI Scene Flow Test***: we compare our method with stereo *(top)* and monocular *(bottom)* scene flow methods. Despite the difficult setting, our fine-tuned model demonstrates encouraging results in real-time.

Our self-supervised learning approach *(Self-Mono-SF)* is outperformed only by Mono-SF (Brickwedde et al., 2019), which is a semi-supervised method using pseudo labels, semantic instance knowledge, and an additional dataset (Cityscapes (Cordts et al., 2016)). However, our method runs more than two orders of magnitude faster. We also provide the accuracy of our fine-tuned model *(Self-Mono-SF-ft)* on the training set for reference.



(a) Input images (b) Monocular depth (c) Optical flow (d) 3D visualization of scene flow

Figure 6.7: **Qualitative results of our monocular scene flow results (Self-Mono-SF-ft) on KITTI 2015 Scene Flow Test**: each scene shows *(a)* two input images, *(b)* monocular depth, *(c)* optical flow, and *(d)* a 3D visualization of estimated depth, overlayed with the reference image, and colored with the (x, z)-coordinates of the 3D scene flow using the standard optical flow color coding.

Table 6.5 shows the comparison with stereo and monocular scene flow methods on the KITTI Scene Flow 2015 benchmark. Fig. 6.7 provides a visualization. Our semi-supervised fine-tuning further improves the accuracy, going toward that of Mono-SF (Brickwedde et al., 2019), but with a more than 400× faster run-time. For further accuracy improvements, *e. g.* rigidity refinement (Jiang et al., 2019; Liu et al., 2019a), exploiting an external dataset (Cordts et al., 2016) for pre-training, or pseudo ground truth (Brickwedde et al., 2019) can be applied on top of our self-supervised learning and semi-supervised fine-tuning pipeline without affecting run-time.

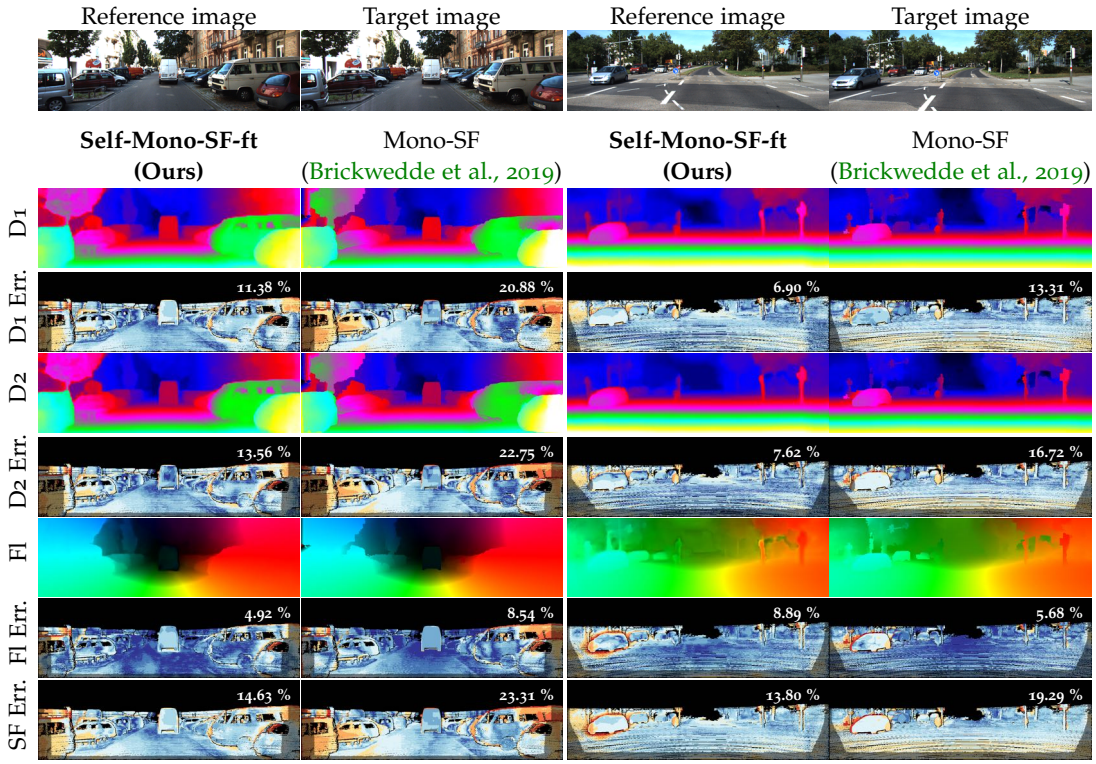


Figure 6.8: **Some successful cases and qualitative comparison with the state of the art on the KITTI 2015 Scene Flow public benchmark (Menze et al., 2015b; 2018).** In the first row, we show two input images, the reference and target image. From the second to the last row, we give a qualitative comparison with Mono-SF (Brickwedde et al., 2019): the disparity map of the reference image *(D1)* with its error map *(D1 Err.)*, disparity estimation at the target image mapped into the reference frame *(D2)* along with its error map *(D2 Err.)*, optical flow *(Fl)* with its error map *(Fl Err.)*, and the scene flow error map *(SF Err.)*. The outlier rates are overlayed on each error map.

6.3.4 *Qualitative Comparison*

We provide some qualitative examples of our monocular scene flow estimation by comparing with the state-of-the-art Mono-SF method (Brickwedde et al., 2019), which uses an integrated pipeline of CNNs and an energy-based model. Figs. 6.8 and 6.9 show successful qualitative results as well as some failure cases of our fine-tuned model on the KITTI 2015 Scene Flow public benchmark (Menze et al., 2015b; 2018), respectively.

In Fig. 6.8, our model outputs more accurate disparity and optical flow estimation results than Mono-SF (Brickwedde et al., 2019) without using an explicit planar surface representation or a rigid motion assumption, which would be beneficial for achieving better accuracy on the KITTI 2015 Scene Flow public benchmark.

Fig. 6.9, in contrast, shows some of the failure cases, where our model outputs less accurate results for scene flow estimation than Mono-SF (Brickwedde et al., 2019). Although our model can estimate optical flow with an accuracy comparable to Mono-

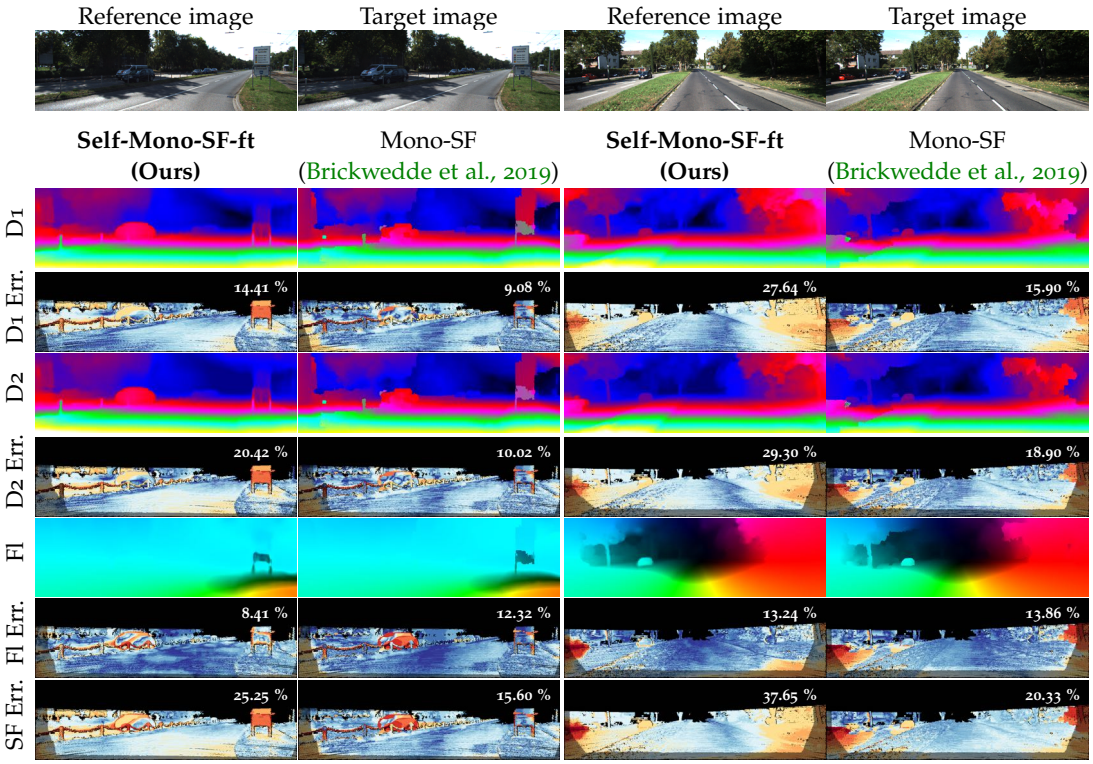


Figure 6.9: **Failure cases and qualitative comparison with the state of the art on the KITTI 2015 Scene Flow public benchmark (Menze et al., 2015b; 2018).** In the first row, we show two input images, the reference and target image. From the second to the last row, we give a qualitative comparison with Mono-SF (Brickwedde et al., 2019): the disparity map of the reference image *(D1)* with its error map *(D1 Err.)*, disparity estimation at the target image mapped into the reference frame *(D2)* with its error map *(D2 Err.)*, optical flow *(Fl)* with its error map *(Fl Err.)*, and the scene flow error map *(SF Err.)*. The outlier rates are overlayed on each error map.

SF, inaccurate disparity estimation eventually leads to less accurate scene flow. The gap in terms of the disparity accuracy of ours *vs*. Mono-SF (Brickwedde et al., 2019) can be explained by the fact that Mono-SF exploits over 20 000 instances of pseudo ground-truth depth data to train their monocular depth model, while our method uses only 200 images for fine-tuning.

6.3.5 *Monocular depth and optical flow*

We also provide a comparison to unsupervised multi-task CNN approaches (Chen et al., 2019; Liu et al., 2019a; Luo et al., 2019; Ranjan et al., 2019; Yang et al., 2018; Yin and Shi, 2018; Zou et al., 2018b) regarding the accuracy of depth and optical flow. We do not report methods that use extra datasets (*e. g.*, the Cityscapes dataset (Cordts et al., 2016)) for pre-training or online fine-tuning (Chen et al., 2019), which is known to give an accuracy boost.

Split	Method	(lower is better) Abs Rel	Sq Rel	RMSE	RMSE log	(higher is better) $\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
KITTI	DF-Net (Zou et al., 2018b)	0.150	1.124	5.507	0.223	0.806	0.933	0.973
	EPC§ (Yang et al., 2018)	0.109	1.004	6.232	0.203	0.853	0.937	0.975
	Liu et al. (2019a)§	0.108	1.020	5.528	0.195	0.863	0.948	0.980
	Self-Mono-SF (Ours)§	**0.106**	**0.888**	**4.853**	**0.175**	**0.879**	**0.965**	**0.987**
Eigen	GeoNet (Yin and Shi, 2018)	0.155	1.296	5.857	0.233	0.793	0.931	0.973
	CC (Ranjan et al., 2019)	0.140	1.070	5.326	0.217	0.826	0.941	0.975
	GLNet(-ref.) (Chen et al., 2019)	0.135	1.070	5.230	0.210	0.841	0.948	**0.980**
	EPC§ (Yang et al., 2018)	0.127	1.239	6.247	0.214	0.847	0.926	0.969
	EPC++§ (Luo et al., 2019)	0.127	**0.936**	5.008	0.209	0.841	0.946	0.979
	Self-Mono-SF (Ours)§	**0.125**	0.978	**4.877**	**0.208**	**0.851**	**0.950**	0.978

Table 6.6: **Monocular depth comparison**: our method demonstrates superior accuracy on the *KITTI split* and competitive accuracy on the *Eigen split* compared to all published multi-task methods. §method using stereo sequences for training.

	Method	Train EPE	Train Fl-all	Test Fl-all
Stereo	Lai et al. (2019)	7.13	27.13	–
	Lee et al. (2019)	8.74	20.88	–
	UnOS (Wang et al., 2019b)	**5.58**	–	**18.00**
Monocular	GeoNet (Yin and Shi, 2018)	10.81	–	–
	DF-Net (Zou et al., 2018b)	8.98	26.01	25.70
	GLNet (Chen et al., 2019)	8.35	–	–
	EPC§ (Yang et al., 2018)	–	25.74	–
	EPC++§ (Luo et al., 2019)	**5.43**	**19.64**	**20.52**
	Liu et al. (2019a)§	5.74	–	–
	Self-Mono-SF (Ours)§	7.51	23.49	23.54

Table 6.7: **Optical flow estimation on the *KITTI split***: our method demonstrates comparable accuracy to both monocular and stereo-based multi-task methods.

For monocular depth estimation in Table 6.6, our monocular scene flow method outperforms all published multi-task methods on the *KITTI Split* (Godard et al., 2017) and demonstrates competitive accuracy on the *Eigen split* (Eigen et al., 2014). Note that some of the methods (Ranjan et al., 2019; Yin and Shi, 2018; Zou et al., 2018b) use *ground truth* to correctly scale their predictions at test time, which gives them an unfair advantage, but are still outperformed by ours.

For optical flow estimation in Table 6.7, our method demonstrates comparable accuracy to existing state-of-the-art monocular (Chen et al., 2019; Yang et al., 2018; Yin and Shi, 2018; Zou et al., 2018b) and stereo methods (Lai et al., 2019; Lee et al., 2019), in part outperforming them.

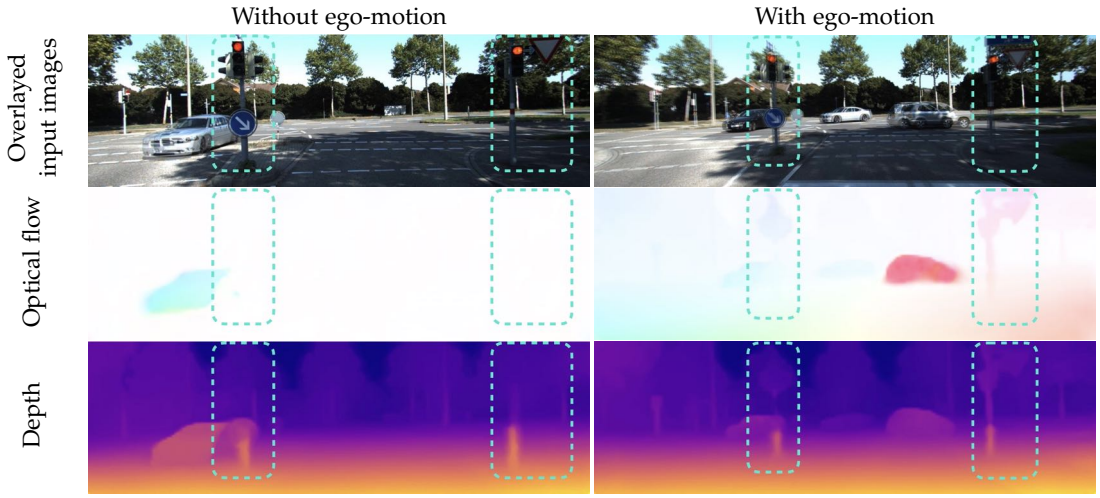


Figure 6.10: **Qualitative examples in the presence of ego-motion:** the left column visualizes overlayed input images, output optical flow and depth when there is no camera ego-motion. The right column demonstrates the outputs when there exists camera ego-motion. Without camera ego-motion, our model relies learned prior knowledge for the depth estimation. In the presence of ego-motion, our model additionally exploits the two-view geometry cue for the depth estimation.

One reason why our flow accuracy may not surpass all previous methods is that we use a 3D scene flow regularizer and not a 2D optical flow regularizer. This is consistent with our goal of estimating 3D scene flow, but it is known that using a regularizer in the target space is critical for achieving the best accuracy (Vogel et al., 2015). While our choice of 3D regularizer is not ideal for optical flow estimation, its benefits manifest in 3D. For example, while we do not outperform EPC++ (Luo et al., 2019) in terms of 2D flow accuracy, we clearly surpass it in terms of scene flow accuracy (see Table 6.4). Consequently, our approach is not only the first CNN approach to monocular scene flow estimation that directly predicts the 3D scene flow, but also outperforms existing multi-task CNNs.

6.3.6 *Qualitative examples on the presence of ego-motion*

Finally, we provide an evidence on how our model perceives depth from two temporally consecutive images. Fig. 6.10 visualizes the two cases where there exists camera ego-motion or not. When the camera is static, our model relies on learned prior knowledge for the depth estimation, similar to conventional monocular depth methods. Our model, unfortunately, fails to properly estimate depth on traffic lights (highlighted with color) in the static scene. In the presence of ego-motion, on the other hand, our model is able to estimate depth on the traffic lights. This suggests that our model additionally exploits the motion cue (*i. e.*, two-view geometry) for the depth estimation as well as learned prior knowledge.

6.4 DISCUSSION

In this chapter, we proposed a CNN-based monocular scene flow estimation approach that jointly estimates depth and 3D scene flow, as our last joint objective. A crucial feature is our single joint decoder for depth and scene flow, which allows overcoming the limitations of existing multi-task approaches such as complex training schedules or lacking occlusion handling. We take a self-supervised approach, where our 3D loss function and occlusion reasoning significantly improve the accuracy. Moreover, we show that a suitable augmentation scheme is critical for competitive accuracy. Our model achieves state-of-the-art scene flow accuracy among un-/self-supervised monocular methods, and our semi-supervised fine-tuned model approaches the accuracy of the best monocular scene flow method to date while being orders of magnitude faster. With competitive accuracy and real-time performance, our method provides a solid foundation for CNN-based monocular scene flow estimation as well as follow-up work.

7

CONCLUSION AND OUTLOOK

In this dissertation, we proposed how to jointly formulate multiple tasks for scene understanding and what kind of benefits can be obtained from the joint estimation. We demonstrated that the joint formulation can provide additional evidence cues to each other (*e. g.* optical flow and semantic segmentation), resolve ambiguities in a chicken-and-egg relationship (*e. g.* optical flow and occlusion), and even simplify multiple tasks (*e. g.* depth and motion via monocular scene flow estimation), which overall improves the accuracy over individual formulations. We conclude the dissertation by summarizing our contributions and presenting future perspecitives.

7.1 CONTRIBUTIONS

Bridging optical flow and semantic segmentation. Though the two tasks are not directly related, we demonstrated that each task can successfully benefits the other in the temporal domain. In Chapter 3, we proposed a piece-wise optical flow model as a baseline model which itself already achieves competitive results. Then from bottom-up semantic cues, we embed semantic information through label consistency and epipolar constraints for static objects in which motion should follow the camera ego-motion. Our experiment on temporally consistent semantic segmentation demonstrates the benefit of our approach by reducing false positives and flickering.

Joint optical flow and occlusion estimation. In Chapter 4, we addressed the chicken-and-egg problem that optical flow and occlusion present, and proposed a symmetrical method that jointly estimates bi-directional optical flow and occlusion maps for each view. Unlike most of the previous methods that consider occlusion as outliers, we formulate to jointly estimate them altogether and demonstrated significant optical flow accuracy improvement especially in occluded areas. On top of a piecewise rigid motion model, we exploit both forward-backward consistency of the flow as well as occlusion-disocclusion symmetry. For the challenging KITTI benchmark at the time of submission, we report leading results even without employing any semantic knowledge or learning of appearance descriptors.

Further in Chapter 5, we demonstrated that optical flow and occlusion can also successfully exploit their relationship in CNN. We proposed an Iterative Residual Refinement (IRR) scheme based on weight sharing for optical flow backbone networks, additionally estimating bi-directional flow and occlusion jointly. This is inspired

by classical energy-based methods that iteratively and residually update previous estimates using a single optimizer. Application of the scheme on top of the two representative backbone networks, FlowNet (Dosovitskiy et al., 2015) and PWC-Net (Sun et al., 2018), significantly improves flow accuracy with a better generalization while even reducing the number of network parameters in case of PWC-Net (Sun et al., 2018). Joint estimation of occlusion with optical flow brings accuracy gains on both tasks and set the state of the art on public benchmark datasets at the time of publication.

Monocular scene flow estimation. In Chapter 6, we proposed a CNN-based monocular scene flow approach that jointly predicts 3D scene flow and depth from two temporally consecutive monocular images. Based on PWC-Net (Sun et al., 2018), we use a single joint decoder that outputs the joint objectives from a classical optical flow cost volume. This simple architecture design choice allows overcoming the limitations of existing multi-task approaches that require complex training schedules due to having multiple decoders. We take a self-supervised approach, where our 3D loss function and occlusion reasoning significantly improve the accuracy. This further allows our model to exploit a large amount of unlabeled data without concerns about the shortage of 3D data annotation. Our model achieves state-of-the-art scene flow accuracy among un-/self-supervised monocular methods, and our semi-supervised fine-tuned model approaches the accuracy of the best monocular scene flow method while being orders of magnitude faster. Furthermore, our model demonstrates competitive accuracy on monocular depth and optical flow subtasks in 2D.

7.2 FUTURE PERSPECTIVES

We successfully demonstrated the advantages of joint formulations, substantially improving accuracy on each task; yet, there still exist many open challenges. In this section, we discuss existing limitations of our proposed methods and future perspectives for further improvement.

Multiple frame estimation. Our methods assume that only two temporally consecutive monocular images are given as input. This is a challenging yet minimal setup to demonstrate the ideas. One straightforward way to extend our methods is to exploit more than two frames. Given multiple temporally consecutive frames, one can always utilize the temporal coherence assumption in the temporal domain, which has been widely demonstrated in the optical flow literature (Janai et al., 2017; Kennedy and Taylor, 2015; Neoral et al., 2018; Ren et al., 2019; Werlberger et al., 2009). Given the assumption, the multi-frame setup can output temporally consistent estimation, which is more robust to outliers or heavy occlusions that the two-frame setup cannot effectively handle. Not only for motion estimation, semantic segmentation (Ding et al., 2020) and depth estimation (Godard et al., 2019) can also benefit from the multi-frame formulation by utilizing low-level image evidence cues from more than two frames or encouraging temporal consistency among temporally neighboring estimates.

Modeling camera motion with instance knowledge. For the monocular scene flow approach in Chapter 6, we do not estimate camera ego-motion unlike the previous approaches (Luo et al., 2019; Ranjan et al., 2019; Yin and Shi, 2018), mainly for the simplicity of the pipeline. Having an extra decoder for the camera motion often requires complex training schedules as well as complicated loss designs for moving objects, and thus increases complexity of methods.

That said, camera ego-motion estimation does help for outdoor scenarios, which mostly consist of static objects. For static objects, their 3D motion relative to the camera is equal to the inverse of the ego-motion of the camera, and thus ego-motion estimation effectively models their motion as well as depth. Yet, motion estimation of independently moving instances that do not follow the camera motion still remains one of the drawbacks of this approach. Utilization of given semantic/instance segmentation (Gordon et al., 2019; Lee et al., 2021) for moving objects can help solve this issue; however, it requires an off-the-shelf method for the additional input, which sometimes does not generalize well on different domains and thus may limit the accuracy of the method. As a future direction, if it is possible to jointly segment out moving instances and accurately model their motion in a self-supervised manner, this can potentially resolve the current limitation and improve the overall accuracy.

Unification of all contributions. In each chapter, we presented each specific case of joint formulations and investigated detailed technical designs for the joint estimation. Yet, our ultimate goal is to formulate and leverage the tasks all together simultaneously. In line with this, several studies (Jiang et al., 2019; Tosi et al., 2020) already demonstrated CNN-based methods that learn from data in an end-to-end manner; however, the lack of interpretability and explainability remains a critical concern for its safe utility in the autonomous navigation. To improve, future research on estimating uncertainty of each output, dependency, or causality between tasks is warranted.

Combination of learning and non-learning modules. Although CNN-based approaches have been leading the current trend these days due to their advantages of fast runtime and great accuracy, non-learning approaches (*e. g.* our approaches in Chapters 3 and 4) still exhibit the benefit of relatively better generalization due to a lower model capacity that yields lower variance on results. This benefit can resolve the main limitation of learning-based approaches that they overfit on the training domain.

In order to benefit from both worlds, a complementary approach can certainly be an interesting direction to pursue. For example, in Chapters 3 and 4, we demonstrate that superpixel can provide informative cues on object boundaries and simplify a per-pixel regression task into a piece-wise regression task. Instead of learning to directly regress per-pixel optical flow from data, learning to over-segment images and to output parameterized motion would simplify the problem, make the learning process easier and less overfit.

Improvement in self-supervised learning. In Chapter 6, we demonstrated learning to estimate 3D motion and depth from purely unlabeled data. Self-supervised learning could be one of the most promising directions to pursue, as it can utilize a large amount unlabeled data on target domains directly. However, several challenges persist, such as lower accuracy than supervised methods in general, a time-consuming process of designing proxy tasks and losses, hyper-parameters to tune, etc.

One viable solution is semi-supervised (Lai et al., 2017) or active learning, which exploits existing annotated data or requires a less data annotation cost on top of self-supervised learning. Active learning reduces the amount of annotation cost substantially, by learning to estimate uncertainty of the current state of output and telling which data should be labeled to reduce the uncertainty.

Or, unsupervised domain adaptation (Tonioni et al., 2019; Zou et al., 2018a) can also overcome such a challenge. It transfers knowledge that is learned in the training domain with a sufficient amount of annotation (*e. g.*, synthetic domain), to the target domain (*e. g.*, real-world domain) where the labeled data is not available or difficult to obtain.

Current proxy loss function designs are based on the brightness constancy assumption, which is often violated in several scenarios besides occlusion, such as specular reflection and illumination changes. Modeling to handle such scenarios in the loss function designs or architecture designs can further advance current methods.

Another direction is to substitute a hand-crafted proxy loss with a feature-metric loss. It has been widely known that careful design choices on the proxy loss matter for better accuracy (Jonschkowski et al., 2020), which also means that the accuracy can be bounded by its proxy loss design. To overcome this, Im et al. (2020) and Shu et al. (2020) presented a feature-metric loss for self-supervised optical flow or monocular depth tasks. Given the view synthesis as a proxy task, they penalize a distance of corresponding pixels in a high-dimensional learned feature space instead of an image space where most of the current works do. As the feature space can have higher representation capacity, it can be more discriminative and informative than image intensity if features are properly learned to behave so. We expect that learning losses to learn can overcome the limitation of the current hand-crafted proxy loss.

A

SUPPLEMENTAL MATERIAL FOR EXPLOITING SYMMETRIES IN JOINT OPTICAL FLOW AND OCCLUSION ESTIMATION

Preface. We here provide additional details for Chapter 4, on the data term, an analysis of the optimizer, an accuracy analysis in occluded regions, and details on the processing time.

A.1 DETAILS ON THE DATA TERM

In Eqs. (4.2b) and (4.2c) in Section 4.2.2, the function $\rho_{\mathrm{D}}(\mathbf{p}, \mathbf{H}_{s_\mathbf{p}})$ measures the photometric error between a pixel $\mathbf{p}$ and its corresponding pixel $\mathbf{H}_{s_\mathbf{p}}\mathbf{p}$ in the other frame. For example, given a homography motion $\mathbf{H}^f_{s_\mathbf{p}}$, the data cost for pixel $\mathbf{p}$ in I^t is given as

$$\rho^f_{\mathrm{D}}(\mathbf{p}, \mathbf{H}^f_{s_\mathbf{p}}) = \min\left\{\rho_l\big(\phi(\mathbf{p}, \mathbf{H}^f_{s_\mathbf{p}})\big), \tau_{\mathrm{D}}\right\} \tag{A.1a}$$

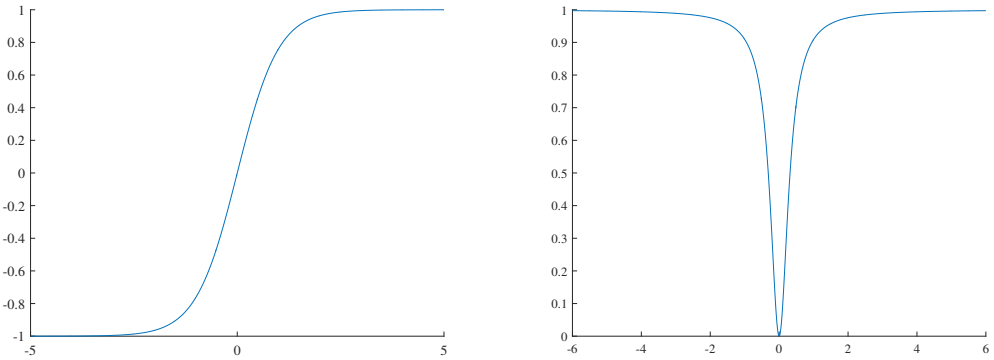
with

$$\begin{aligned}\phi(\mathbf{p}, \mathbf{H}^f_{s_\mathbf{p}}) = \alpha_D \sum_{\mathbf{y}\in\{-3,\dots,3\}^2} f\Bigg(&\underbrace{T\big(I^t(\mathbf{p}+\mathbf{y}) - I^t(\mathbf{p})\big)}_{\text{ternary value at } \mathbf{p} \text{ in } I^t} \\ &- \underbrace{T\big(I^{t+1}(\mathbf{H}^f_{s_\mathbf{p}}(\mathbf{p}+\mathbf{y})) - I^{t+1}(\mathbf{H}^f_{s_\mathbf{p}}\mathbf{p})\big)}_{\text{ternary value at } \mathbf{H}^f_{s_\mathbf{p}}\mathbf{p} \text{ in } I^{t+1}}\Bigg) \\ +(1-\alpha_D)\big|&\underbrace{\nabla I^{t+1}(\mathbf{H}^f_{s_\mathbf{p}}\mathbf{p}) - \nabla I^t(\mathbf{p})}_{\text{gradient constancy penalty}}\big|,\end{aligned} \tag{A.1b}$$

which is the weighted sum of the ternary transform and gradient constancy penalty.

Deviations are penalized by a Lorentzian penalty $\rho_l(x) = \alpha_l \log((1+x^2)/2\sigma_l^2)$, truncated at τ_{D}. The idea behind function $\phi(\mathbf{p}, \mathbf{H}^f_{s_\mathbf{p}})$ is to calculate the Hamming distance of two 7×7 ternary patches, one around pixel $\mathbf{p}$ in I^t and one around the corresponding pixel $\mathbf{H}^f_{s_\mathbf{p}}\mathbf{p}$ in I^{t+1}. Unlike the conventional ternary transform (Stein, 2004), we use a continuous variant. Specifically, we relax the definition of the Hamming distance and adopt the sigmoid function

$$T(x) = \frac{2}{1+\exp(-\sigma_T x)} - 1 = \frac{1-\exp(-\sigma_T x)}{1+\exp(-\sigma_T x)} \tag{A.2}$$



(a) (b)

Figure A.1: *(a)* Sigmoid function. *(b)* Geman-McClure function.

instead of a true ternary value, and use the Geman-McClure function (Black and Rangarajan, 1996) to score the differences in the ternary signature between the patches:

$$f(x) = \frac{x^2}{(\sigma_f + x^2)}. \tag{A.3}$$

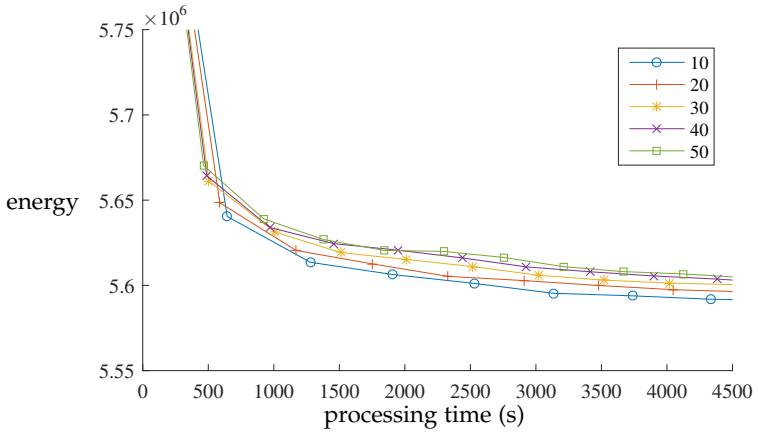
As shown in Figs. A.1a and A.1b, these continuous functions approximate the conventional discrete setting, but they assess subtle brightness variations more naturally when their input is near zero. In other words, they are still robust, but less brittle than the original Hamming-based definition.

Furthermore, when calculating the ternary value at point $\mathbf{H}^f_{s_\mathbf{p}}\mathbf{p}$ in the other frame, we calculate it not on the conventional ternary patch that is centered at the transformed point, but on a transformed patch. Eq. (A.1b) precisely expresses how to calculate the ternary value on the warped patch (*i. e.*, referring the intensity at $\mathbf{H}^f_{s_\mathbf{p}}(\mathbf{p}+\mathbf{y})$ instead of $\mathbf{H}^f_{s_\mathbf{p}}\mathbf{p}+\mathbf{y}$). Similar to a classical iterative-warping scheme, this strategy yields a more comprehensive data cost that is invariant to local shape deformation caused by the motion.

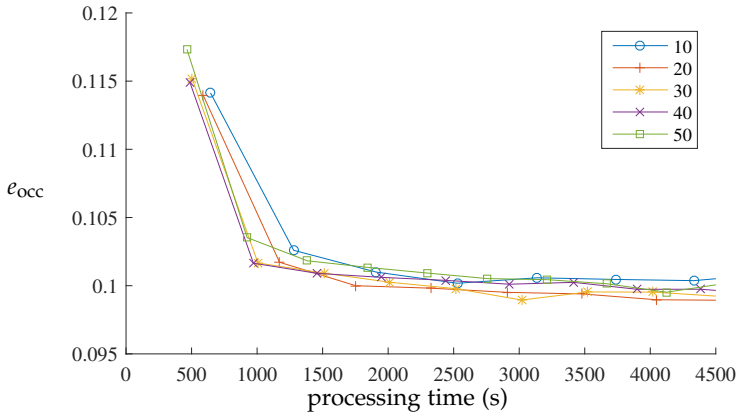
We observe that the two practices above increase the flow accuracy. Table A.1 compares the flow accuracy of three different ways of calculating the ternary value: *(i)* our standard implementation including both the continuous ternary variant and the patch

	Non-occluded pixels			All pixels		
Method	Fl-bg	Fl-fg	Fl-all	Fl-bg	Fl-fg	Fl-all
standard	**6.52%**	**11.72%**	**7.41%**	**9.26%**	**13.94%**	**9.98%**
discrete	7.11%	12.26%	7.99%	9.88%	14.57%	10.60%
w/o transformation	7.29%	12.35%	8.16%	10.41%	14.80%	11.08%

Table A.1: Evaluation of different methods for computing the ternary census on the KITTI training set. See text for details.



(a) Overall energy depending on the number of superpixels in each subgraph



(b) The estimated flow error rates depending on the number of superpixels in each subgraph

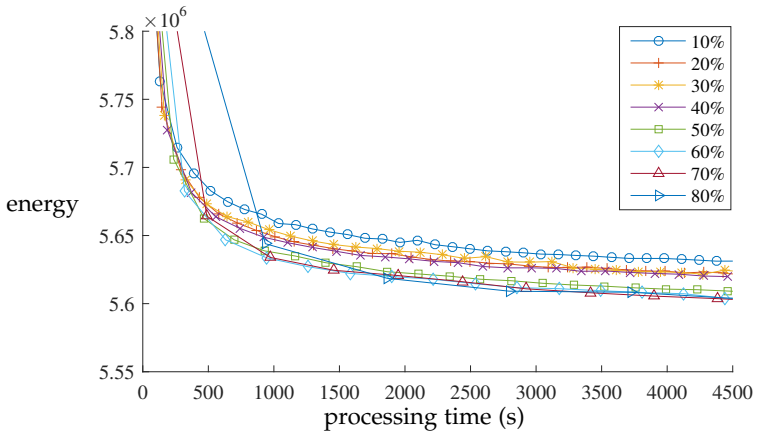
Figure A.2: Overall energy and the estimated flow error rates depending on the number of superpixels in each subgraph.

transformation (*standard*), *(ii)* the standard implementation with the conventional discrete setting of the ternary transform (*discrete*) but with patch transformation, *(iii)* the standard implementation without the patch transformation (*w/o transformation*) but with the continuous variant.

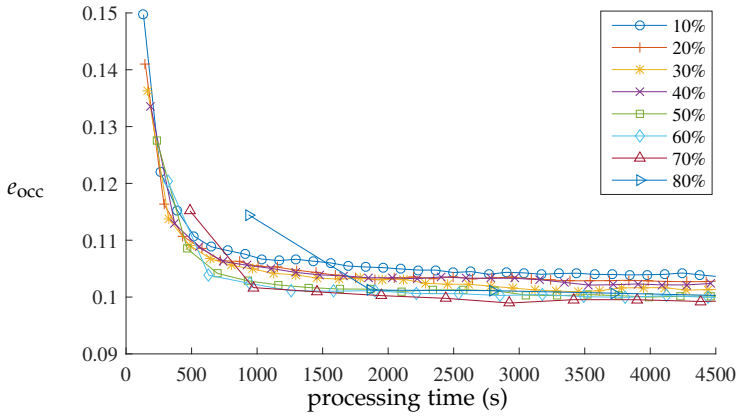
As presumed, using the continuous ternary variant and the transformed ternary patch both yield better accuracy by reducing the number of flow errors by about 5.85 % and 9.93 % respectively. This experiment has been conducted on the KITTI Optical Flow 2015 training set.

A.2 ANALYSIS OF THE OPTIMIZER

As discussed in Section 4.2.3, we first collect a set of proposals when assigning homography motions for each superpixel, and sequentially run expansion moves on each subgraph of superpixels to allow for an efficient optimization. We assemble a set of subgraphs in a way that each subgraph consists of neighboring 30 superpixels with



(a) Overall energy depending on the overlap setting



(b) The estimated flow error rates depending on the overlap setting

Figure A.3: Overall energy and the estimated flow error rates depending on the overlap setting.

70 % overlap between each other. We empirically found that this is an advantageous setting in our problem.

Choosing the number of superpixels in each subgraph affects both flow accuracy and energy at convergence. When the number of superpixels is high, proposals can be propagated into broader regions, but the algorithm has a smaller chance of finding locally optimal homography motions, which results in slower convergence. On the other hand, when the number of superpixels is low, locally optimal motions can lower the energy level more quickly, but the optimization can get stuck in local optima as it propagates labels only in small regions, which eventually leads to higher flow error rates.

Figure A.2a and Fig. A.2b demonstrate the energy and the flow error rate (on KITTI 2015 training), respectively, versus the processing time, depending on the number of superpixels in each subgraph. Each dot on a graph represents an iteration step. These two figures illustrate the trade-off described above. We found that having 30 superpixels for each subgraph yields the lowest flow error rates.

Choosing the size of overlapping regions between subgraphs also incurs a trade-off: When the size is getting bigger, the proposals can be propagated more effectively between subgraphs, which helps finding lower energy solutions in fewer iterations. However, it requires more processing time per iteration because the size of the subgraphs is increased. When the size of overlaps gets smaller, on the other hand, less processing time per iteration is needed, but the optimizer propagates proposals through subgraphs less effectively, leading to more iterations being required.

Figure A.3a and Fig. A.3b demonstrate the energy and the flow error rate (on KITTI 2015 training), respectively, versus processing time, depending on the overlap size between subgraphs. As in Fig. A.3a, if the overlap size is more than 50 %, the energy is converging to a lower value, but with a similar speed. Figure A.3b demonstrates that having 70 % of overlap between subgraphs yields the lowest flow error rates in the same processing time. However, please note that the flow accuracy differences between the settings are not very significant (< 5 %).

A.3 PERFORMANCE IN OCCLUDED REGIONS

We analyze the flow estimation accuracy of top-performing algorithms including ours especially in occluded regions on the KITTI Optical Flow 2015 benchmark (Menze et al., 2015b; 2018). Unlike the MPI Sintel Flow Dataset (Butler et al., 2012), the KITTI benchmark does not explicitly provide the statistics for occluded areas. Thus, we indirectly deduce them.

To that end, let us define the variables $n_{\text{all}}, n_{\text{noc}}, n_{\text{occ}}, e_{\text{all}}, e_{\text{noc}}$, and e_{occ} as follows:

- n_{all} : no. of all pixels considered in evaluation
- n_{noc} : no. of non-occluded pixels
- n_{occ} : no. of occluded pixels
- e_{all} : no. of all pixels with an incorrect flow estimate
- e_{noc} : no. of non-occluded pixels with an incorrect flow estimate
- e_{occ} : no. of occluded pixels with an incorrect flow estimate.

Then, the following equations naturally hold:

$$n_{\text{all}} = n_{\text{noc}} + n_{\text{occ}} \tag{A.4a}$$

$$e_{\text{all}} = e_{\text{noc}} + e_{\text{occ}}. \tag{A.4b}$$

We are interested in estimating the flow error rate in occluded areas, $e_{\text{occ}}/n_{\text{occ}}$. From Eqs. (A.4a) and (A.4b) we have

$$\frac{e_{\text{occ}}}{n_{\text{occ}}} = \frac{e_{\text{all}} - e_{\text{noc}}}{n_{\text{all}} - n_{\text{noc}}} = \frac{\frac{e_{\text{all}}}{n_{\text{all}}} - \frac{e_{\text{noc}}}{n_{\text{all}}}}{1 - \frac{n_{\text{noc}}}{n_{\text{all}}}} = \frac{\frac{e_{\text{all}}}{n_{\text{all}}} - \frac{e_{\text{noc}}}{n_{\text{noc}}}\frac{n_{\text{noc}}}{n_{\text{all}}}}{1 - \frac{n_{\text{noc}}}{n_{\text{all}}}}. \tag{A.5}$$

Given that we do not know the ratio of non-occluded pixels, we substitute $n_{\text{noc}}/n_{\text{all}}$ with α in Eq. (A.5) and obtain

$$\frac{e_{\text{occ}}}{n_{\text{occ}}} = \frac{1}{1-\alpha}\frac{e_{\text{all}}}{n_{\text{all}}} - \frac{\alpha}{1-\alpha}\frac{e_{\text{noc}}}{n_{\text{noc}}}, \tag{A.6}$$

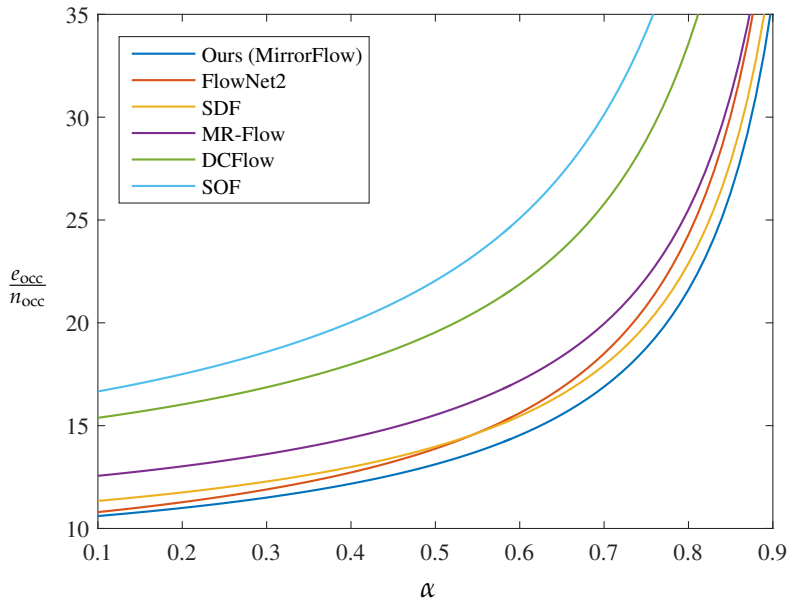


Figure A.4: The estimated flow error rates of top-performing algorithms in occluded regions with respect to $\alpha = n_{noc}/n_{all}$.

where the values e_{all}/n_{all} and e_{noc}/n_{noc} are the flow error rates on all pixels and non-occluded pixels, respectively, which can be found in Table 4.1. Therefore, we can indirectly infer the flow error rate in occluded areas based on the statistics from Table 4.1 and in terms of $\alpha = n_{noc}/n_{all}$, which denotes the (unknown) ratio of the number of non-occluded pixels to that of all pixels that are considered in the evaluation.

In Fig. A.4 we now plot the estimated flow error rates of top-performing algorithms in occluded regions by varying the unknown ratio α. We observe that our MirrorFlow approach consistently shows the lowest error rates among the top-performing algorithms regardless of values of the ratio α. Considering that the ratio α for the KITTI 2015 training set is $\alpha = 0.8635$, we can confidently infer that our method demon-

Method	Fl-all in occluded pixels (estimates)
Ours (MirrorFlow)	**28.19** %
SDF (Bai et al., 2016)	29.80 %
FlowNet2 (Ilg et al., 2017)	32.36 %
MR-Flow (Wulff et al., 2017)	36.23 %
DCFlow (Xu et al., 2017)	44.47 %
SOF (Sevilla-Lara et al., 2016)	54.33 %

Table A.2: Estimated flow errors for occluded pixels (when $\alpha = 0.8635$). Our method demonstrates the lowest error among all published two-frame methods on the KITTI benchmark.

strates the lowest optical flow error among the top-performing algorithms on the KITTI benchmark. Table A.2 gives the estimated results assuming the same α as on the training set.

A.4 PROCESSING TIME

For processing a 1226×370 image, the algorithm takes around 40 minutes on a single core until the accuracy no longer increases (tested on Intel Xeon CPU E5-2650 2.20GHz). Yet, the algorithm can be easily parallelized because the local subgraphs that do not overlap with each other can be processed at the same time (Taniai et al., 2017). Using 4 cores, the runtime decreases down to 11 minutes.

The main bottleneck is calculating the ternary transform in the data term. We calculate the ternary census on transformed patches, which needs to be done for every different homography motion. When just using a plain data term (penalizing intensity + gradient differences), the runtime is only 4 minutes. CNN-based learned descriptors also have the potential to lead to a speedup as a future work.

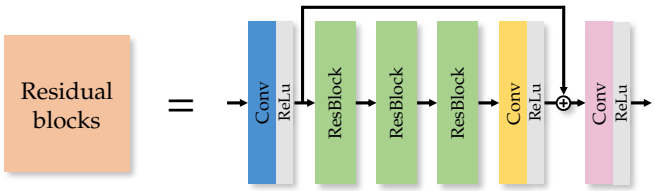
B

SUPPLEMENTAL MATERIAL FOR ITERATIVE RESIDUAL REFINEMENT FOR JOINT OPTICAL FLOW AND OCCLUSION ESTIMATION

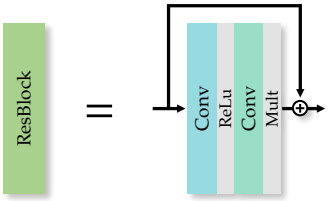
Preface. Here, we provide additional details on Chapter 5, especially on the occlusion upsampling layer and more qualitative examples on the ablation study.

B.1 DETAILS ON THE OCCLUSION UPSAMPLING LAYER

In the occlusion upsampling layer shown in Fig. 5.7, the *residual blocks* (Lim et al., 2017) are fed a set of feature maps as input and output residual occlusion estimates to refine the upscaled occlusion map from the previous level. Fig. B.1 shows the details of the *residual blocks*. As shown in Fig. B.1a, the subnetwork consists of 3 residual blocks (*i. e.* 3 *ResBlock*s) with 3 convolution layers. One *ResBlock* consists of *Conv+ReLu+Conv+Mult* operations as shown in Fig. B.1b, *cf*. Lim et al. (2017). This sequence of 3 *ResBlock*s with one convolution layer afterwards estimates the residuals over one convolution output of the input feature maps, and the final convolution layer of the *residual blocks* outputs the residual occlusion. The number of channels for



(a) Residual blocks subnetwork.



(b) One residual block.

Figure B.1: **Residual blocks in the upsampling layer**: *(a)* The *residual blocks* consist of 3 weight-shared *ResBlock*s with 3 convolution layers. *(b)* One *ResBlock* consists of *Conv+ReLu+Conv+Mult* operations (Lim et al., 2017).

all convolution layers here is 32, except for the final convolution layer, which has only 1 channel for the occlusion output.

We use weight sharing also on the upsampling layers between bi-directional estimations and between pyramid levels or iteration steps. Furthermore, the *ResBlock*s in Fig. B.1a also share their weights, which is different from Lim et al. (2017), where they are not shared. With this efficient weight-sharing scheme, the occlusion upsampling layer improves the occlusion accuracy by 2.99% on the training domain (*i. e.* the FlyingChairsOcc dataset) and 4.08% across datasets (*i. e.* Sintel) with only adding 0.031 M parameters.

B.2 ADDITIONAL QUALITATIVE EXAMPLES

B.2.1 *Occlusion upsampling layer*

Fig. B.2 provides qualitative examples of occlusion estimation and demonstrates the advantage of using the occlusion upsampling layer. The models used here are trained on the FlyingChairsOcc dataset only (no fine-tuning on the FlyingThings3D-subset dataset or Sintel) and tested on Sintel Train Clean. The occlusion upsampling layer enhances the occlusion estimates to be much sharper along motion boundaries and refines coarse estimates. Also, the upsampling layer further detects thinly-shaped occlusions that were not detected at the quarter resolution. Unlike optical flow, where a quarter resolution estimate is largely sufficient, we can see from these qualitative examples that estimating occlusions up to the original resolution is very critical for yielding high accuracy.

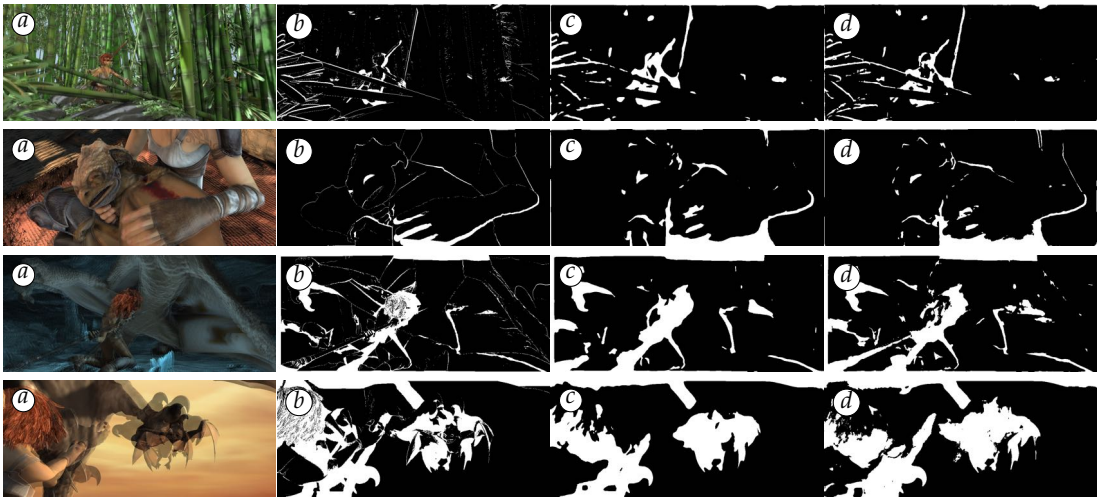


Figure B.2: **Qualitative examples of using the occlusion upsampling layer**: *(a)* overlapped input images, *(b)* occlusion ground truth, *(c)* without using the occlusion upsampling layer, and *(d)* with using the occlusion upsampling layer. The occlusion upsampling layer makes occlusion estimates much sharper along motion boundaries and detects additional thinly-shaped occlusions.

Figure B.3: **More qualitative examples from the ablation study on PWC-Net**: *(a)* overlapped input images, *(b)* the original PWC-Net (Sun et al., 2018), *(c)* PWC-Net with Bi, *(d)* PWC-Net with Occ, *(e)* PWC-Net with Bi-Occ, *(f)* optical flow ground truth, *(g)* PWC-Net with IRR, *(h)* PWC-Net with Occ-IRR, *(i)* PWC-Net with Bi-Occ-IRR, and *(j)* our full model (i.e. IRR-PWC). Our full model significantly improves flow estimation over the original PWC-Net with fewer missing details and clearer motion boundaries. Note that there are gradual improvements when combining several of the proposed components.

B.2.2 *Ablation study on PWC-Net*

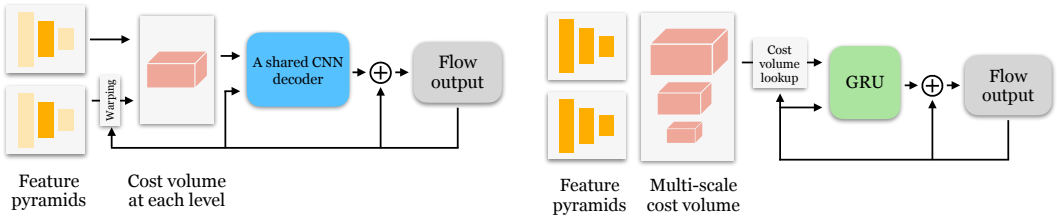
In addition to Fig. 5.11 in the main chapter, we here give more qualitative examples for the ablation study. In Fig. B.3, all models are also trained on the FlyingChairsOcc dataset and tested on Sintel Train Clean. Our proposed schemes significantly improve the accuracy over the baseline model (*i. e.* PWC-Net (Sun et al., 2018)), yielding better generalization across datasets.

B.3 COMPARISON WITH RAFT

RAFT (Teed and Deng, 2020) also follows similar ideas to our idea on the iterative residual refinement, but technical designs are slightly different. Table B.1 and Fig. B.4 summarize the main difference on the technical designs of both methods.

Cost volume. Our method computes a cost volume at each pyramid level. In contrast, RAFT pre-computes a multi-scale 4D correlation volume for all possible pairs of pixels.

Warping method. Our method backward-warps feature maps at the target view before the cost volume construction. On the other hand, RAFT performs lookups on the pre-computed cost volume, which can be equivalent to a forward-warping operation.



(a) IRR-PWC (Ours). (b) RAFT (Teed and Deng, 2020).

Figure B.4: **Network architecture comparison with our model and RAFT**: *(a)* our method constructs a cost volume at each pyramid level and uses a shared CNN decoder for the update. *(b)* RAFT pre-computes an all-pair multi-scale cost volume and performs lookups at each iteration step using a GRU to update flow.

Model	**IRR-PWC (Ours)**	RAFT (Teed and Deng, 2020)
Cost volume	Cost-volume at each pyramid level	All-pair multi-scale cost volume (pre-computation)
Warping method	Backward warping	Forward warping (cost volume lookup)
Update module	A shared CNN decoder	GRU
Update operation	Residual update along pyramid levels	Residual update at a single resolution

Table B.1: **Technical design comparison between our method and RAFT**: our method and RAFT are based on the same core idea, iterative and residual refinement, but details on cost volume construction and the way of residual update are different.

Update module. We use a shared CNN decoder for all pyramid levels. RAFT uses a GRU unit.

Update operation. Our method residually updates along the pyramid levels. Thus, the number of iteration steps is the same as the number of pyramid levels. In contrast, RAFT recurrently updates flow at a single resolution using the GRU unit. Thus, the number of iteration steps can be flexible.

C

SUPPLEMENTAL MATERIAL FOR SELF-SUPERVISED MONOCULAR SCENE FLOW ESTIMATION

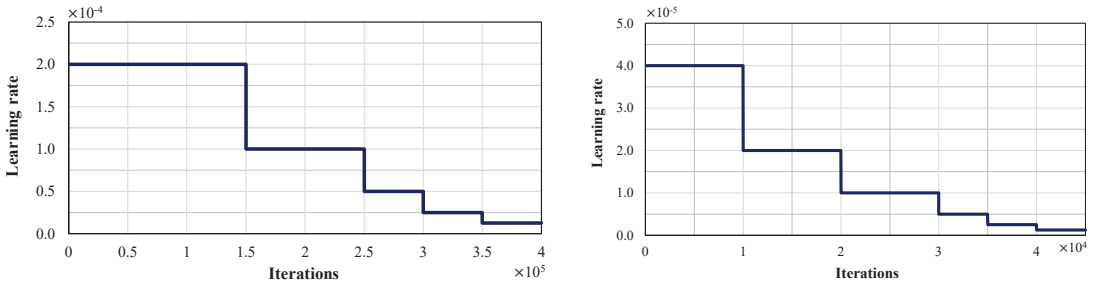
Preface. In this supplementary material, we provide further details for Chapter 6 on the learning rate schedules, data augmentation, and the hyper-parameter settings. Afterwards, we provide a more comprehensive study of the decoder design and qualitative examples for the loss ablation study.

C.1 LEARNING RATE SCHEDULE

Fig. C.1 illustrates the learning rate schedules for both self-supervised learning and semi-supervised fine-tuning. When first training our model in a self-supervised manner for 400k iterations, the initial learning rate starts from 2×10^{-4} and is halved at 150k, 250k, 300k, and 350k iteration steps. When fine-tuning in a semi-supervised manner afterwards, the training schedule consists of 45k iterations; the initial learning rate starts from 4×10^{-5} and is halved at 10k, 20k, 30k, 35k, and 40k iteration steps.

C.2 DETAILS ON DATA AUGMENTATION

As discussed in Section 6.3.1, we perform photometric and geometric augmentations at training time. Here we provide more details on our augmentation setup for both self-supervised training and semi-supervised fine-tuning.



(a) Learning rate schedule for self-supervised learning.

(b) Learning rate schedule for fine-tuning.

Figure C.1: **Learning rate schedules** for *(a)* self-supervised learning and *(b)* semi-supervised fine-tuning.

Augmentations for self-supervised training. We apply photometric augmentations with 50% probability. Specifically, we adopt random gamma adjustments, uniformly sampled from $[0.8, 1.2]$, brightness changes with a multiplication factor that is uniformly sampled in $[0.5, 2.0]$, and random color changes with a multiplication factor that is uniformly sampled in $[0.8, 1.2]$ for each color channel.

For geometric augmentations, we first randomly crop the input images with a random scale factor uniformly sampled in $[93\%, 100\%]$ and apply random translations uniformly sampled from $[-3.5\%, 3.5\%]$ *w. r. t.* the input image size. Then we resize the cropped image to 256×832 pixels as in previous work (Lee et al., 2019; Liu et al., 2019a; Luo et al., 2019; Ranjan et al., 2019; Yang et al., 2018). We also apply a horizontal flip (Lai et al., 2019; Lee et al., 2019; Liu et al., 2019a; Wang et al., 2019b) with 50% probability. Because the geometric augmentations have an effect on the camera intrinsics, we adjust the intrinsic camera matrix accordingly by calculating the corresponding camera center and focal length of each augmented image. At testing time, we only resize the input image to 256×832 pixels without photometric augmentation.

Augmentations for semi-supervised fine-tuning. Likewise, we also apply the same photometric augmentations with 50% probability. For geometric augmentations, we only apply random cropping without scaling and then resize to 256×832 pixels. Not performing scaling is to avoid changes to the ground truth, which may happen if zooming and interpolating the sparse ground truth. The crop size $s \cdot h_0 \times s \cdot w_0$ is determined by the cropping factor s that is uniformly sampled in $[94\%, 100\%]$, where h_0 and w_0 is height and width of the original input resolution. At testing time, the same augmentation scheme as during self-supervised training applies: resizing the input images to 256×832 pixels without photometric augmentation. However, we note that better augmentation protocols can likely be discovered with further investigation (Bar-Haim and Wolf, 2020).

C.3 HYPER-PARAMETER SETTINGS

Our self-supervised proxy loss in Eq. (6.1) in Section 6.2.4 has a total of 6 hyper-parameters, which could make it difficult to achieve satisfactory results without careful tuning. In this section, we thus discuss how we choose the hyper-parameters and provide an analysis on how sensitive the scene flow accuracy is depending on the hyper-parameter choices.

First, as discussed in Section 6.3.1, the balancing weight λ_{sf} between the two joint tasks in Eq. (6.1) is dynamically determined to make the loss of the scene flow and disparity be equal in every iteration (Hur and Roth, 2019). For the disparity loss, we simply adopt the same hyper-parameters (*i. e.*, $\lambda_{\text{d\_sm}}$, α, and β in Eqs. (6.2), (6.3b) and (6.4), respectively) as in previous work (Godard et al., 2017), which leaves only two hyper-parameters, $\lambda_{\text{sf\_sm}}$ and $\lambda_{\text{sf\_pt}}$, to tune in the scene flow loss, Eq. (6.5). We perform grid search on the two parameters.

Table C.1 gives the grid search results regarding the two hyper-parameters, reporting the accuracy for monocular depth, optical flow, and scene flow. In the upper

		Depth	Flow	Scene Flow			
λ_{sf_sm}	λ_{sf_pt}	Abs Rel	EPE	D1-all	D2-all	Fl-all	SF-all
1	0.005	**0.104**	7.118	**30.50**	51.48	**22.32**	62.97
	0.05	0.107	**7.057**	32.56	49.45	22.33	61.27
	0.1	0.109	7.319	33.65	**35.57**	22.58	**47.46**
	0.5	0.117	8.259	33.91	36.24	25.18	48.72
10	0.005	**0.105**	**6.934**	**31.18**	52.29	**22.15**	63.47
	0.2	0.108	7.421	31.37	**34.39**	22.73	**46.08**
	0.3	0.110	7.379	31.91	34.42	23.79	47.10
	0.4	0.113	7.773	32.79	35.53	23.98	47.63
200	0.005	**0.103**	**6.883**	**30.48**	50.05	**22.65**	61.47
	0.1	0.108	7.525	31.49	46.50	23.38	59.17
	0.2	0.107	7.197	31.40	**34.75**	23.02	**46.95**
	0.4	0.114	7.435	33.35	35.56	24.30	48.25
0.1	0.005	0.106	6.839	31.47	52.20	22.39	63.70
1		**0.104**	7.118	**30.50**	51.48	22.32	62.97
10		0.105	6.934	31.18	52.29	**22.15**	63.47
100		0.105	**6.723**	31.15	**51.05**	22.18	**62.55**
1	0.2	0.109	**7.118**	31.81	34.95	23.01	46.82
10		0.108	7.421	31.37	**34.39**	**22.73**	**46.08**
100		0.108	7.386	**31.05**	34.95	22.88	47.08
200		**0.107**	7.197	31.40	34.75	23.02	46.95
10	0.4	0.113	7.773	32.79	35.53	23.98	47.63
100		**0.111**	**7.365**	32.97	**34.63**	**23.92**	**47.29**
200		0.114	7.435	33.35	35.56	24.30	48.25
300		0.112	7.833	**31.97**	35.20	25.39	48.48

Table C.1: **Grid search results on the two hyper-parameters, λ_{sf_sm} and λ_{sf_pt}** based on the accuracy of monocular depth, optical flow, and scene flow. The 3D point reconstruction parameter λ_{sf_pt} contributes to more accurate disparity on the target frame, *D2-all*, yielding more accurate scene flow *SF-all* in the end. The overall results are not so sensitive to the choice of the smoothness parameter λ_{sf_sm}.

half of the table, we fix the smoothness parameter λ_{sf_sm} and control the 3D point reconstruction loss parameter λ_{sf_pt} to see its effect on the accuracy. The bottom half of the table is set up the other way around. Note that the lower the better for all metrics.

We find that λ_{sf_pt} is important for best scene flow accuracy, specifically settings that yield accurate disparity information on the target frame, *D2-all*. This observation follows our design of the 3D point reconstruction loss, which penalizes the 3D distance between corresponding points, encouraging more accurate 3D scene flow in 3D space. However, as a trade-off, having a higher value of λ_{sf_pt} leads to lower

accuracy for 2D estimation, *i.e.* of depth and optical flow. On the other hand, we find that the parameter for the 3D smoothness loss, $\lambda_{\text{sf\_sm}}$, does not strongly affect the accuracy in general. That is, once $\lambda_{\text{sf\_pt}}$ is in the right range, the results are not particularly sensitive to the parameter choice.

C.4 IN-DEPTH ANALYSIS OF THE DECODER DESIGN

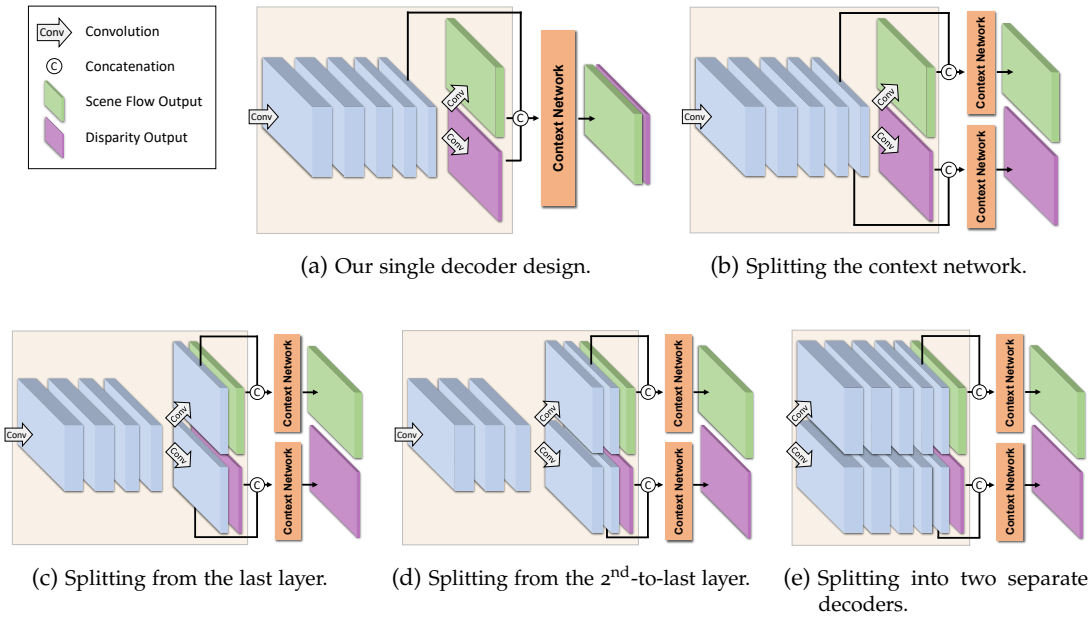
With the decoder ablation study in Table 6.3 of the main paper, we demonstrate that having separate decoders for disparity and scene flow yields instable, unbalanced outputs in contrast to having our proposed single decoder design. For a more comprehensive analysis, we conduct an empirical study by gradually splitting the decoder consisting of 5 convolution layers and studying the behavior of the networks for each configuration. Our backbone network, PWC-Net (Sun et al., 2018), has context networks at the end of the decoder, which are fed the output and the last feature map from the decoder as input and perform post-processing for better accuracy. In our splitting study, we also separate the context networks for each separated decoder so that the two decoders at the end of the networks do not share information.

Fig. C.2 illustrates each configuration. From our single decoder design in Fig. C.2a, we first split the context network for disparity and scene flow respectively, as shown in Fig. C.2b. Then, we begin to split the decoder from the last convolution layer (*i.e.*, Fig. C.2c), the 2$^{\text{nd}}$-to-last layer (*i.e.*, Fig. C.2d), and so on until eventually completely splitting into two separate decoders (*i.e.*, Fig. C.2e). To ensure the same network capacity, we adjust the number of filters so that all configurations have network parameter numbers in a similar range. All configurations are trained on the *KITTI Split* of KITTI raw (Geiger et al., 2013) in our self-supervised manner.

Table C.2 shows the disparity, optical flow, and scene flow accuracy of each configuration on *KITTI Scene Flow Training* (Menze et al., 2015b; 2018). We first observe that splitting the context network yields a significant 32.73% decrease in scene flow accuracy (*i.e.*, SF-all), which mainly stems from the less accurate disparity estimates (*i.e.*, D1-all and D2-all) although the optical flow accuracy remains almost the same. This provides an important outlook: given the same optical flow accuracy, the scene flow accuracy depends crucially on how well one can decompose the optical flow cost volume into depth and scene flow, where using the single decoder model works better. When further splitting the decoder starting from the last convolution layer, the networks *(i)* cannot be trained stably anymore, *(ii)* output trivial solutions for the disparity, and *(iii)* even decrease the optical flow accuracy. This observation again confirms the benefits of using our proposed single decoder design in terms of both accuracy and training stability.

C.5 QUALITATIVE ANALYSIS OF LOSS ABLATION STUDY

Table 6.2 in the main paper provides an ablation study of our self-supervised proxy loss. For better understanding of how each loss term affects the results, we provide qualitative examples of disparity, optical flow, and scene flow estimation. Fig. C.3



(a) Our single decoder design.

(b) Splitting the context network.

(c) Splitting from the last layer.

(d) Splitting from the 2nd-to-last layer.

(e) Splitting into two separate decoders.

Figure C.2: **Gradually splitting the single decoder into two separate decoders**: we gradually split the single decoder *(a)* by first splitting the context network *(b)*, and then splitting from the last layer of the decoder *(c)*, the 2nd-to-last layer *(d)*, and so on until completely splitting into two separate decoders *(e)*. For ease of visualization, we omit showing the *convolution* operation between the neighboring feature maps in the decoder.

Configuration	D1-all	D2-all	Fl-all	SF-all
Single decoder	**31.25**	**34.86**	**23.49**	**47.05**
Splitting the context network	44.19	45.02	23.51	62.45
Splitting at the last layer	100	97.22	26.46	100
Splitting at the 2nd-to-last layer	100	97.22	26.39	100
Splitting at the 3rd-to-last layer	100	97.22	26.94	100
Splitting at the 4th-to-last layer	100	97.22	28.68	100
Splitting into two separate decoders	100	97.22	27.63	100

Table C.2: **Scene flow accuracy of each decoder configuration**: splitting the context network already decreases the scene flow accuracy by 32.73%. Further splitting the decoder yields training instability with trivial solutions for the disparity output.

displays the results for each loss configuration: *(a)* the basic loss where only the brightness and smoothness terms are active; *(b)* with occlusion handling, which discards occluded pixels in the loss; *(c)* with the 3D point reconstruction loss; and *(d)* the full loss. Each configuration is trained in the proposed self-supervised manner using the KITTI *Split* and evaluated on KITTI *Scene Flow Training* (Menze et al., 2015b; 2018).

Without the 3D point reconstruction loss for scene flow (*i. e.*, columns *(a)* and *(b)* in Fig. C.3), the networks output inaccurate disparity information for the target frame *(D2)* especially in the road area, which yields inaccurate scene flow results *(SF1)* in the end. Applying the 3D point reconstruction loss but without occlusion handling (*i. e.*, column *(c)* in Fig. C.3) results in inaccurate estimates and some artifacts appearing on out-of-bound pixels, still leading to an unsatisfactory final scene flow accuracy. These artifacts happen when the 3D point reconstruction loss tries to minimize the 3D Euclidean distance between incorrect pixel correspondences, such as for occlusions or out-of-bound pixels. Discarding those occluded regions in the proxy loss eventually yields better estimates in the occluded region as well.

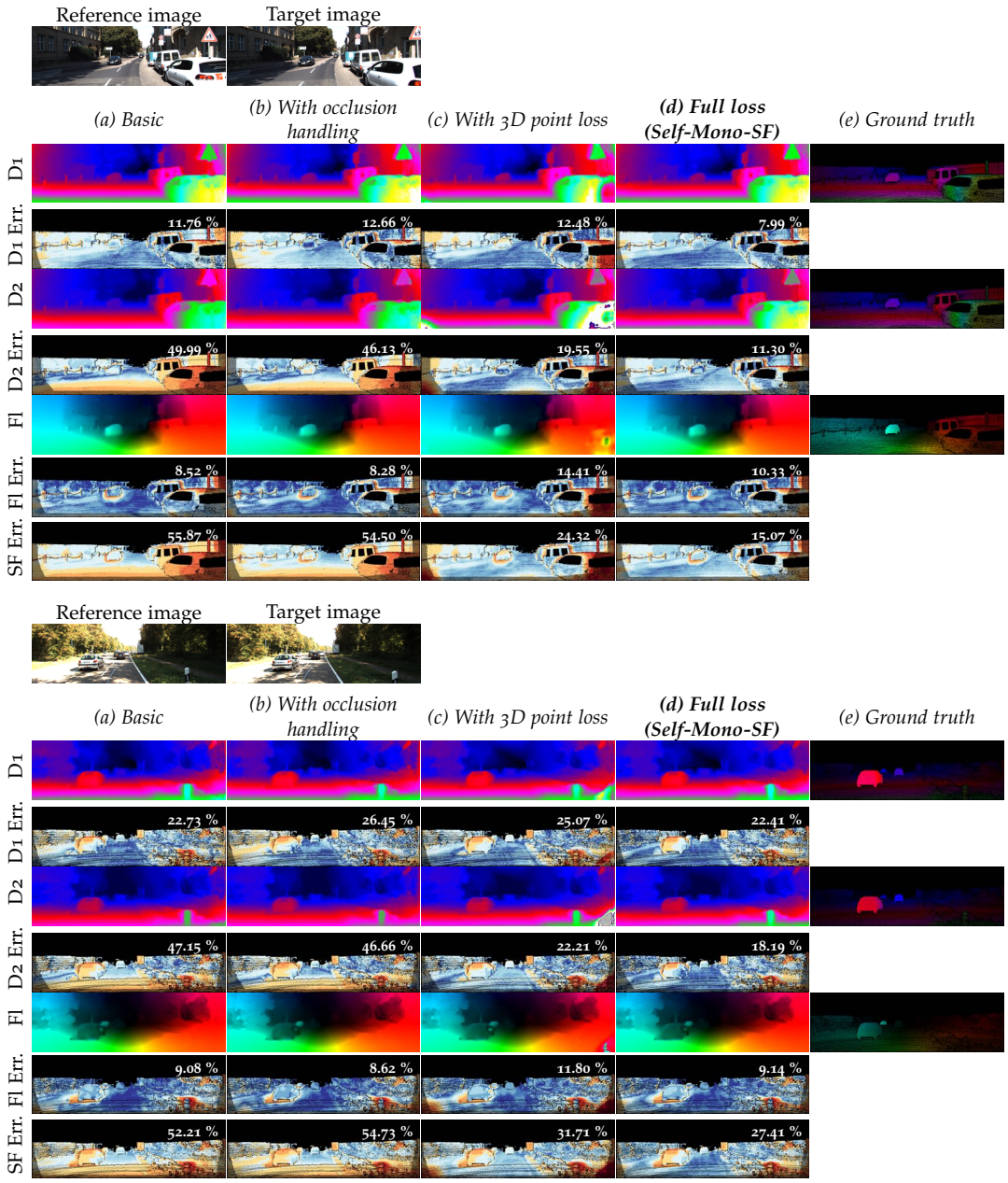


Figure C.3: **Qualitative examples on the loss ablation study**. For each scene in the first row we show two input images, the reference and the target image. From the second to the last row, we show a qualitative comparison of each loss configuration: *(a)* basic loss, *(b)* with occlusion handling, *(c)* with 3D point reconstruction loss, and the *(d)* our full loss. Each row visualizes the disparity map of the reference image *(D1)* with its error map *(D1 Err.)*, disparity estimation at the target image mapped into the reference frame *(D2)* along with its error map *(D2 Err.)*, optical flow *(Fl)* with its error map *(Fl Err.)*, and the scene flow error map *(SF Err.)*. The outlier rates are overlayed on each error map. The last column shows *(e)* the ground truth for each estimate.

BIBLIOGRAPHY

Ahmadi, Aria and Ioannis Patras (2016). "Unsupervised Convolutional Neural Networks for Motion Estimation." In: *Proceedings of the IEEE International Conference on Image Processing*, pp. 1629–1633.

Alvarez León, Luis Miguel, Julio Esclarín Monreal, Martin Lefébure, and Javier Sánchez (Sept. 1999). "A PDE Model for Computing the Optical Flow." In: *Proc. XVI Congreso de Ecuaciones Diferenciales y Aplicaciones*. Las Palmas de Gran Canaria, Spain, pp. 1349–1356.

Alvarez, Luis, Rachid Deriche, Théodore Papadopoulo, and Javier Sánchez Pérez (2007). "Symmetrical Dense Optical Flow Estimation with Occlusions Detection." In: *International Journal of Computer Vision* 75.3, pp. 371–385.

Alvarez, Luis, Joachim Weickert, and Javier Sánchez (Aug. 2000). "Reliable Estimation of Dense Optical Flow Fields with Large Displacements." In: *International Journal of Computer Vision* 39.1, pp. 41–56.

Arbeláez, Pablo, Bharath Hariharan, Chunhui Gu, Saurabh Gupta, Lubomir Bourdev, and Jitendra Malik (June 2012). "Semantic Segmentation Using Regions and Parts." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Providence, Rhode Island, pp. 3378–3385.

Aubry, Mathieu, Daniel Maturana, Alexei A. Efros, Bryan C. Russell, and Josef Sivic (June 2014). "Seeing 3D Chairs: Exemplar Part-Based 2D-3D Alignment Using a Large Dataset of CAD Models." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, Ohio, pp. 3762–3769.

Bai, Min, Wenjie Luo, Kaustav Kundu, and Raquel Urtasun (2016). "Exploiting Semantic Information and Deep Matching for Optical Flow." In: *Proceedings of the 14th European Conference on Computer Vision*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Vol. 9910. Lecture Notes in Computer Science. Springer, pp. 154–170.

Bailer, Christian, Bertram Taetz, and Didier Stricker (Dec. 2015). "Flow Fields: Dense Correspondence Fields for Highly Accurate Large Displacement Optical Flow Estimation." In: *Proceedings of the Fifteenth IEEE International Conference on Computer Vision*. Santiago, Chile, pp. 4015–4023.

Bailer, Christian, Kiran Varanasi, and Didier Stricker (July 2017). "CNN-based Patch Matching for Optical Flow with Thresholded Hinge Embedding Loss." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, pp. 2710–2719.

Baker, Simon, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski (Mar. 2011). "A Database and Evaluation Methodology for Optical Flow." In: *International Journal of Computer Vision* 92.1, pp. 1–31.

Ballester, Coloma, Lluís Garrido, Vanel Lazcano, and Vicent Caselles (2012). "A TV-L1 Optical Flow Method with Occlusion Detection." In: *Pattern Recognition, Proceedings*

of the 34th DAGM-Symposium. Ed. by A. Pinz, T. Pock, H. Bischof, and F. Leberl. Vol. 7476. Lecture Notes in Computer Science. Springer, pp. 31–40.

Bar-Haim, Aviram and Lior Wolf (June 2020). "ScopeFlow: Dynamic Scene Scoping for Optical Flow." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Virtual.

Barnes, Connelly, Eli Shechtman, Dan B. Goldman, and Adam Finkelstein (2010). "The Generalized PatchMatch Correspondence Algorithm." In: *Proceedings of the 11th European Conference on Computer Vision*. Ed. by K. Daniilidis, P. Maragos, and N. Paragios. Vol. 6313. Lecture Notes in Computer Science. Springer, pp. 29–43.

Behl, Aseem, Omid Hosseini Jafari, Siva Karthik Mustikovela, Hassan Abu Alhaija, Carsten Rother, and Andreas Geiger (Oct. 2017). "Bounding Boxes, Segmentations and Object Coordinates: How Important is Recognition for 3D Scene Flow Estimation in Autonomous Driving Scenarios?" In: *Proceedings of the Sixteenth IEEE International Conference on Computer Vision*. Venice, Italy, pp. 2574–2583.

Behl, Aseem, Despoina Paschalidou, Simon Donné, and Andreas Geiger (June 2019). "PointFlowNet: Learning Representations for Rigid Motion Estimation from Point Clouds." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7962–7971.

Besse, Frederic, Carsten Rother, Andrew Fitzgibbon, and Jan Kautz (Oct. 2013). "PMBP: PatchMatch Belief Propagation for Correspondence Field Estimation." In: *International Journal of Computer Vision* 110.1, pp. 2–13.

Black, Michael J. and P. Anandan (Jan. 1996). "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields." In: *Computer Vision and Image Understanding* 63.1, pp. 75–104.

Black, Michael J. and Anand Rangarajan (July 1996). "On the Unification of Line Processes, Outlier Rejection, and Robust Statistics with Applications in Early Vision." In: *International Journal of Computer Vision* 19.1, pp. 57–91.

Boykov, Yuri and Vladimir Kolmogorov (Sept. 2004). "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.9, pp. 1124–1137.

Boykov, Yuri, Olga Veksler, and Ramin Zabih (June 1998). "Markov Random Fields with Efficient Approximations." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Santa Barabara, California, pp. 648–655.

– (Nov. 2001). "Fast Approximate Energy Minimization via Graph Cuts." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.11, pp. 1222–1239.

Brickwedde, Fabian, Steffen Abraham, and Rudolf Mester (Oct. 2019). "Mono-SF: Multi-View Geometry Meets Single-View Depth for Monocular Scene Flow Estimation of Dynamic Traffic Scenes." In: *Proceedings of the Seventeenth IEEE International Conference on Computer Vision*. Seoul, Korea, pp. 2780–2790.

Brostow, Gabriel J., Jamie Shotton, Julien Fauqueur, and Roberto Cipolla (2008). "Segmentation and Recognition Using Structure from Motion Point Clouds." In: *Proceedings of the Tenth European Conference on Computer Vision*. Vol. 5302. Lecture Notes in Computer Science. Springer, pp. 44–57.

Brox, Thomas, Andrés Bruhn, Nils Papenberg, and Joachim Weickert (2004). "High Accuracy Optical Flow Estimation Based on a Theory for Warping." In: *Proceedings of*

the Eighth European Conference on Computer Vision. Vol. 4. Lecture Notes in Computer Science. Springer, pp. 25–36.

Bruhn, Andrés, Joachim Weickert, and Christoph Schnörr (Feb. 2005). "Lucas/Kanade meets Horn/Schunck: Combining Local and Global Optic Flow Methods." In: *International Journal of Computer Vision* 61.3, pp. 211–231.

Butler, Daniel J., Jonas Wulff, Garrett B. Stanley, and Michael J. Black (2012). "A Naturalistic Open Source Movie for Optical Flow Evaluation." In: *Proceedings of the 12th European Conference on Computer Vision*. Vol. 7575. Lecture Notes in Computer Science. Springer, pp. 611–625.

Charbonnier, Pierre, Laure Blanc-Feéraud, Gilles Aubert, and Michel Barlaud (Nov. 1994). "Two Deterministic Half-Quadratic Regularization Algorithms for Computed Imaging." In: *Proceedings of the IEEE International Conference on Image Processing*. Vol. 2. Austin, Texas, pp. 168–172.

Chen, Albert Y. C. and Jason J. Corso (2011). "Temporally Consistent Multi-Class Video-Object Segmentation with the Video Graph-Shifts Algorithm." In: *IEEE Winter Conference on Applications of Computer Vision*.

Chen, Dongdong, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua (Oct. 2017a). "Coherent Online Video Style Transfer." In: *Proceedings of the Sixteenth IEEE International Conference on Computer Vision*. Venice, Italy, pp. 1114–1123.

Chen, Liang-Chieh, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D. Collins, Ekin D. Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens (2020). "Naive-Student: Leveraging Semi-Supervised Learning in Video Sequences for Urban Scene Segmentation." In: *Proceedings of the 16th European Conference on Computer Vision*. Springer.

Chen, Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille (June 2017b). "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4, pp. 834–848.

Chen, Qifeng and Vladlen Koltun (June 2016). "Full Flow: Optical Flow Estimation by Global Optimization over Regular Grids." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, Nevada, pp. 4706–4714.

Chen, Yuhua, Cordelia Schmid, and Cristian Sminchisescu (Oct. 2019). "Self-Supervised Learning With Geometric Constraints in Monocular Video: Connecting Flow, Depth, and Camera." In: *Proceedings of the Seventeenth IEEE International Conference on Computer Vision*. Seoul, Korea, pp. 7063–7072.

Cheng, Jingchun, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang (Oct. 2017). "SegFlow: Joint Learning for Video Object Segmentation and Optical Flow." In: *Proceedings of the Sixteenth IEEE International Conference on Computer Vision*. Venice, Italy, pp. 686–695.

Cordts, Marius, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele (June 2016). "The Cityscapes Dataset for Semantic Urban Scene Understanding." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, Nevada, pp. 3213–3223.

Dalal, Navneet and Bill Triggs (June 2005). "Histograms of Oriented Gradients for Human Detection." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. San Diego, California, pp. 886–893.

Dijk, Tom van and Guido de Croon (Oct. 2019). "How Do Neural Networks See Depth in Single Images?" In: *Proceedings of the Seventeenth IEEE International Conference on Computer Vision*. Seoul, Korea, pp. 2183–2191.

Ding, Mingyu, Zhe Wang, Bolei Zhou, Jianping Shi, Zhiwu Lu, and Ping Luo (Feb. 2020). "Every Frame Counts: Joint Learning of Video Segmentation and Optical Flow." In: *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*. New York, New York, pp. 10713–10720.

Dosovitskiy, Alexey, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick v. d. Smagt, Daniel Cremers, and Thomas Brox (Dec. 2015). "FlowNet: Learning Optical Flow with Convolutional Networks." In: *Proceedings of the Fifteenth IEEE International Conference on Computer Vision*. Santiago, Chile, pp. 2758–2766.

Drulea, Marius and Sergiu Nedevschi (2013). "Motion Estimation Using the Correlation Transform." In: *IEEE Transactions on Image Processing* 22.8, pp. 3260–3270.

Eigen, David, Christian Puhrsch, and Rob Fergus (2014). "Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network." In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Vol. 27, pp. 2366–2374.

Ellis, Willis D. (1938). *A Source Book of Gestalt Psychology*. London: Kegan Paul, Trench, Trubner & Company.

Facil, Jose M., Benjamin Ummenhofer, Huizhong Zhou, Luis Montesano, Thomas Brox, and Javier Civera (June 2019). "CAM-Convs: Camera-Aware Multi-Scale Convolutions for Single-View Depth." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Long Beach, California, pp. 11826–11835.

Fischler, Martin A. and Robert C. Bolles (1981). "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." In: *Communications of the ACM* 24.6, pp. 381–395.

Floros, Georgios and Bastian Leibe (June 2012). "Joint 2D-3D Temporally Consistent Semantic Segmentation of Street Scenes." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Providence, Rhode Island.

Fortun, Denis, Patrick Bouthemy, and Charles Kervrann (May 2015). "Optical Flow Modeling and Computation: A Survey." In: *Computer Vision and Image Understanding* 134.1, pp. 1–21.

Gadde, Raghudeep, Varun Jampani, and Peter V. Gehler (Oct. 2017). "Semantic Video CNNs through Representation Warping." In: *Proceedings of the Sixteenth IEEE International Conference on Computer Vision*. Venice, Italy, pp. 4463–4472.

Gadot, David and Lior Wolf (June 2016). "PatchBatch: A Batch Augmented Loss for Optical Flow." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, Nevada, pp. 4236–4245.

Geiger, Andreas, Philip Lenz, Christoph Stiller, and Raquel Urtasun (Aug. 2013). "Vision Meets Robotics: The KITTI dataset." In: *International Journal of Robotics Research* 32.11, pp. 1231–1237.

Geiger, Andreas, Philip Lenz, and Raquel Urtasun (June 2012). "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Providence, Rhode Island.

Girshick, Ross B., Jeff Donahue, Trevor Darrell, and Jitendra Malik (Jan. 2016). "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.1, pp. 142–158.

Godard, Clément, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow (Oct. 2019). "Digging into Self-Supervised Monocular Depth Estimation." In: *Proceedings of the Seventeenth IEEE International Conference on Computer Vision*. Seoul, Korea, pp. 3828–3838.

Godard, Clément, Oisin Mac Aodha, and Gabriel J. Brostow (July 2017). "Unsupervised Monocular Depth Estimation with Left-Right Consistency." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, pp. 270–279.

Gordon, Ariel, Hanhan Li, Rico Jonschkowski, and Anelia Angelova (Oct. 2019). "Depth from Videos in the Wild: Unsupervised Monocular Depth Learning from Unknown Cameras." In: *Proceedings of the Seventeenth IEEE International Conference on Computer Vision*. Seoul, Korea, pp. 8977–8986.

Grundmann, Matthias, Vivek Kwatra, Mei Han, and Irfan Essa (June 2010). "Efficient Hierarchical Graph-based video Segmentation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. San Francisco, California.

Gu, Xiuye, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang (June 2019). "HPLFlowNet: Hierarchical Permutohedral Lattice FlowNet for Scene Flow Estimation on Large-scale Point Clouds." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Long Beach, California, pp. 3254–3263.

Guizilini, Vitor, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon (June 2020). "3D Packing for Self-Supervised Monocular Depth Estimation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Virtual, pp. 2485–2494.

Güney, Fatma and Andreas Geiger (2016). "Deep Discrete Flow." In: *Proceedings of the Thirteenth Asian Conference on Computer Vision*. Ed. by S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato. Vol. 10115. Lecture Notes in Computer Science. Springer, pp. 207–224.

Gupta, Saurabh, Ross Girshick, Pablo Arbeláez, and Jitendra Malik (2014). "Learning Rich Features from RGB-D Images for Object Detection and Segmentation." In: *Proceedings of the 13th European Conference on Computer Vision*. Vol. 8695. Lecture Notes in Computer Science. Springer, pp. 345–360.

Hadfield, Simon and Richard Bowden (Nov. 2011). "Kinecting the Dots: Particle Based Scene Flow from Depth Sensors." In: *Proceedings of the Thirteenth IEEE International Conference on Computer Vision*. Barcelona, Spain, pp. 2290–2295.

Hadsell, Raia, Sumit Chopra, and Yann LeCun (June 2006). "Dimensionality Reduction by Learning an Invariant Mapping." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. New York, New York, pp. 1735–1742.

Hafner, David, Oliver Demetz, and Joachim Weickert (2013). "Why is the Census Transform Good for Robust Optic Flow Computation?" In: *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 210–221.

Harley, Adam W., Konstantinos G. Derpanis, and Iasonas Kokkinos (Oct. 2017). "Segmentation-Aware Convolutional Networks Using Local Attention Masks." In: *Proceedings of the Sixteenth IEEE International Conference on Computer Vision*. Venice, Italy, pp. 5048–5057.

Hartley, Richard I. (June 1997). "In Defense of the Eight-Point Algorithm." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.6, pp. 580–593.

Hirschmüller, Heiko (Feb. 2008). "Stereo Processing by Semiglobal Matching and Mutual Information." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2, pp. 328–341.

Hoffman, Judy, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell (July 2018). "CyCADA: Cycle-consistent adversarial domain adaptation." In: *Proceedings of the 35th International Conference on Machine Learning*. Stockholm, Sweden, pp. 1989–1998.

Hofinger, Markus, Samuel Rota Bulò, Lorenzo Porzi, Arno Knapitsch, Thomas Pock, and Peter Kontschieder (2020). "Improving Optical Flow on a Pyramid Level." In: *Proceedings of the 16th European Conference on Computer Vision*. Springer.

Horn, Berthold K. P. and Brian G. Schunck (Aug. 1981). "Determining Optical Flow." In: *Artificial Intelligence* 17.1–3, pp. 185–203.

Hornáček, Michael, Frederic Besse, Jan Kautz, Andrew Fitzgibbon, and Carsten Rother (2014). "Highly Overparameterized Optical Flow Using PatchMatch Belief Propagation." In: *Proceedings of the 13th European Conference on Computer Vision*. Vol. 8691. Lecture Notes in Computer Science. Springer, pp. 220–234.

Hornáček, Michael, Andrew Fitzgibbon, and Carsten Rother (June 2014). "SphereFlow: 6 DoF Scene Flow from RGB-D Pairs." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, Ohio, pp. 3526–3533.

Hu, Junjie, Yan Zhang, and Takayuki Okatani (Oct. 2019). "Visualization of Convolutional Neural Networks for Monocular Depth Estimation." In: *Proceedings of the Seventeenth IEEE International Conference on Computer Vision*. Seoul, Korea, pp. 3869–3878.

Hu, Yinlin, Yunsong Li, and Rui Song (July 2017). "Robust Interpolation of Correspondences for Large Displacement Optical Flow." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, pp. 4791–4799.

Hu, Yinlin, Rui Song, and Yunsong Li (June 2016). "Efficient Coarse-to-Fine PatchMatch for Large Displacement Optical Flow." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, Nevada, pp. 5704–5712.

Huang, Gao, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger (July 2017). "Densely Connected Convolutional Networks." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, pp. 770–778.

Huguet, Frédéric and Frédéric Devernay (Oct. 2007). "A Variational Method for Scene Flow Estimation from Stereo Sequences." In: *Proceedings of the Eleventh IEEE International Conference on Computer Vision*. Rio de Janeiro, Brazil, pp. 1–7.

Hui, Tak-Wai and Chen Change Loy (2020). "LiteFlowNet3: Resolving Correspondence Ambiguity for More Accurate Optical Flow Estimation." In: *Proceedings of the 16th European Conference on Computer Vision*. Springer.

Hui, Tak-Wai, Xiaoou Tang, and Chen Change Loy (June 2018). "LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, Utah, pp. 8981–8989.

Hur, Junhwa and Stefan Roth (2016). "Joint Optical Flow and Temporally Consistent Semantic Segmentation." In: *4th Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving*. Ed. by Gang Hua and Hervé Jégou. Vol. 9913. Lecture Notes in Computer Science. jointly with ECCV 2016. Springer, pp. 163–177.

– (Oct. 2017). "MirrorFlow: Exploiting Symmetries in Joint Optical Flow and Occlusion Estimation." In: *Proceedings of the Sixteenth IEEE International Conference on Computer Vision*. Venice, Italy, pp. 312–321.

– (June 2019). "Iterative Residual Refinement for Joint Optical Flow and Occlusion Estimation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Long Beach, California, pp. 5747–5756.

– (2020a). "Optical Flow Estimation in the Deep Learning Age." In: *Modelling Human Motion*. Ed. by Nicoletta Noceti, Alessandra Sciutti, and Francesco Rea. Springer, pp. 119–140.

– (June 2020b). "Self-Supervised Monocular Scene Flow Estimation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Virtual, pp. 7396–7405.

Ilg, Eddy, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox (July 2017). "FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, pp. 1647–1655.

Ilg, Eddy, Tonmoy Saikia, Margret Keuper, and Thomas Brox (2018). "Occlusions, Motion and Depth Boundaries with a Generic Network for Disparity, Optical Flow or Scene Flow Estimation." In: *Proceedings of the 15th European Conference on Computer Vision*. Ed. by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss. Lecture Notes in Computer Science. Springer, pp. 614–630.

Im, Woobin, Tae-Kyun Kim, and Sung-Eui Yoon (2020). "Unsupervised Learning of Optical Flow with Deep Feature Similarity." In: *Proceedings of the 16th European Conference on Computer Vision*. Springer, pp. 172–188.

Ince, Serdar and Janusz Konrad (Aug. 2008). "Occlusion-Aware Optical Flow Estimation." In: *IEEE Transactions on Image Processing* 17.8, pp. 1443–1451.

Isard, Michael and John MacCormick (2006). "Dense Motion and Disparity Estimation via Loopy Belief Propagation." In: *Proceedings of the Seventh Asian Conference on Computer Vision*. Ed. by P. J. Narayanan, Shree K. Nayar, and Heung-Yeung Shum. Vol. 3852. Lecture Notes in Computer Science. Springer, pp. 32–41.

Izadi, Shahram et al. (2011). "KinectFusion: Real-Time 3D Reconstruction and Interaction Using a Moving Depth Camera." In: *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 559–568.

Jaderberg, Max, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu (2015). "Spatial Transformer Networks." In: *Advances in Neural Information Processing Systems*, pp. 2017–2025.

Janai, Joel, Fatma Güney, Anurag Ranjan, Michael J. Black, and Andreas Geiger (2018). "Unsupervised Learning of Multi-Frame Optical Flow with Occlusions." In: *Proceedings of the 15th European Conference on Computer Vision*. Vol. 11220. Lecture Notes in Computer Science. Springer, pp. 713–731.

Janai, Joel, Fatma Guney, Jonas Wulff, Michael J. Black, and Andreas Geiger (July 2017). "Slow Flow: Exploiting High-Speed Cameras for Accurate and Diverse Optical Flow Reference Data." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1406–1416.

Jiang, Huaizu, Deqing Sun, Varun Jampani, Zhaoyang Lv, Erik Learned-Miller, and Jan Kautz (Oct. 2019). "SENSE: A Shared Encoder Network for Scene-flow Estimation." In: *Proceedings of the Seventeenth IEEE International Conference on Computer Vision*. Seoul, Korea, pp. 3195–3204.

Jiang, Shihao, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley (Oct. 2021). "Learning to estimate hidden motions with global motion aggregation." In: *Proceedings of the Seventeenth IEEE International Conference on Computer Vision*. Virtual, pp. 9772–9781.

Jonschkowski, Rico, Austin Stone, Jonathan T. Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova (2020). "What Matters in Unsupervised Optical Flow." In: *Proceedings of the 16th European Conference on Computer Vision*. Springer.

Kennedy, Ryan and Camillo J. Taylor (2015). "Optical Flow with Geometric Occlusion Estimation and Fusion of Multiple Frames." In: *Proceedings of the 10th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*. Ed. by X.-C. Tai, E. Bae, T.F. Chan, and M. Lysaker. Lecture Notes in Computer Science, pp. 364–377.

Kingma, Diederik P. and Jimmy Lei Ba (May 2015). "Adam: A Method for Stochastic Optimization." In: *Proceedings of the International Conference on Learning Representations*. San Diego, California.

Kitt, Bernd and Henning Lategahn (2012). "Trinocular Optical Flow Estimation for Intelligent Vehicle Applications." In: *Proceedings of the IEEE Conference on Intelligent Transportation Systems*.

Kolmogorov, Vladimir and Ramin Zabih (July 2001). "Computing Visual Correspondence with Occlusions Using Graph Cuts." In: *Proceedings of the Eighth IEEE International Conference on Computer Vision*. Vancouver, British Columbia, Canada, pp. 508–515.

Kolmogorov, Vladmimir and Ramin Zabih (Feb. 2004). "What Energy Functions can be Minimized via Graph Cuts?" In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.2, pp. 147–159.

Kontschieder, Peter, Samuel Rota Bulo, Horst Bischof, and Marcello Pelillo (Nov. 2011). "Structured Class-Labels in Random Forests for Semantic Image Labelling." In: *Proceedings of the Thirteenth IEEE International Conference on Computer Vision*. Barcelona, Spain, pp. 2190–2197.

Krähenbühl, Philipp and Vladlen Koltun (2012). "Efficient Nonlocal Regularization for Optical Flow." In: *Proceedings of the 12th European Conference on Computer Vision*. Vol. 7572. Lecture Notes in Computer Science. Springer, pp. 356–369.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks." In: *Advances in Neural Information Processing Systems*. Vol. 25, pp. 1097–1105.

Kundu, Abhijit, Yin Li, Frank Dellaert, Fuxin Li, and James M. Rehg (2014). "Joint Semantic Segmentation and 3D Reconstruction from Monocular Video." In: *Proceedings of the 13th European Conference on Computer Vision*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Lecture Notes in Computer Science. Springer, pp. 703–718.

Kundu, Abhijit, Vibhav Vineet, and Vladlen Koltun (June 2016). "Feature Space Optimization for Semantic Video Segmentation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, Nevada.

Lai, Hsueh-Ying, Yi-Hsuan Tsai, and Wei-Chen Chiu (June 2019). "Bridging Stereo Matching and Optical Flow via Spatiotemporal Correspondence." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Long Beach, California, pp. 1890–1899.

Lai, Wei-Sheng, Jia-Bin Huang, and Ming-Hsuan Yang (2017). "Semi-Supervised Learning for Optical Flow with Generative Adversarial Networks." In: *Advances in Neural Information Processing Systems*, pp. 354–364.

Laina, Iro, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab (2016). "Deeper Depth Prediction with Fully Convolutional Residual Networks." In: *Proceedings of the International Conference on 3D Vision*, pp. 239–248.

Lee, Seokju, Sunghoon Im, Stephen Lin, and In So Kweon (Nov. 2019). "Learning Residual Flow as Dynamic Motion from Stereo Videos." In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Macau, China, pp. 1180–1186.

– (Feb. 2021). "Learning Monocular Depth in Dynamic Scenes via Instance-Aware Projection Consistency." In: *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*. A virtual conference.

Lempitsky, Victor, Stefan Roth, and Carsten Rother (June 2008). "FusionFlow: Discrete-Continuous Optimization for Optical Flow Estimation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Anchorage, Alaska.

Leordeanu, Marius, Andrei Zanfir, and Cristian Sminchisescu (Dec. 2013). "Locally Affine Sparse-to-Dense Matching for Motion and Occlusion Estimation." In: *Proceedings of the Fourteenth IEEE International Conference on Computer Vision*. Sydney, Australia, pp. 1721–1728.

Li, Stan Z. (1994). "Markov Random Field Models in Computer Vision." In: *Proceedings of the Third European Conference on Computer Vision*. Ed. by J.-O. Eklundh. Lecture Notes in Computer Science. Springer, pp. 361–370.

Li, Yu, Dongbo Min, Michael S. Brown, Minh N. Do, and Jiangbo Lu (Dec. 2015). "SPM-BP: Sped-up PatchMatch Belief Propagation for Continuous MRFs." In: *Proceedings of the Fifteenth IEEE International Conference on Computer Vision*. Santiago, Chile, pp. 4006–4014.

Li, Yu, Dongbo Min, Minh N. Do, and Jiangbo Lu (2016). "Fast Guided Global Interpolation for Depth and Motion." In: *Proceedings of the 14th European Conference on Computer Vision*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Vol. 9907. Lecture Notes in Computer Science. Springer, pp. 717–733.

Lim, Bee, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee (July 2017). "Enhanced Deep Residual Networks for Single Image Super-Resolution." In: *CVPR Workshops*. Honolulu, Hawaii, pp. 1132–1140.

Lin, Guosheng, Chunhua Shen, Anton v. d. Hengel, and Ian Reid (June 2016). "Efficient Piecewise Training of Deep Structured Models for Semantic Segmentation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, Nevada, pp. 3194–3203.

Liu, Liang, Guangyao Zhai, Wenlong Ye, and Yong Liu (Aug. 2019a). "Unsupervised Learning of Scene Flow Estimation Fusing with Local Rigidity." In: *Proceedings of the International Joint Conference on Artificial Intelligence*. Macao, China, pp. 876–882.

Liu, Liang, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang (June 2020). "Learning by Analogy: Reliable Supervision from Transformations for Unsupervised Optical Flow Estimation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Virtual, pp. 6489–6498.

Liu, Pengpeng, Irwin King, Michael R Lyu, and Jia Xu (Jan. 2019b). "DDFlow: Learning Optical Flow with Unlabeled Data Distillation." In: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*. Honolulu, Hawaii, pp. 8770–8777.

Liu, Pengpeng, Michael Lyu, Irwin King, and Jia Xu (June 2019c). "SelFlow: Self-Supervised Learning of Optical Flow." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Long Beach, California, pp. 4571–4580.

Liu, Xingyu, Charles R. Qi, and Leonidas J. Guibas (June 2019d). "FlowNet3D: Learning Scene Flow in 3D Point Clouds." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Long Beach, California, pp. 529–537.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (June 2015). "Fully Convolutional Networks for Semantic Segmentation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Boston, Massachusetts, pp. 3431–3440.

Lourakis, M.I.A. (2011). *fundest: A C/C++ Library for Robust, Non-linear Fundamental Matrix Estimation*. http://www.ics.forth.gr/~lourakis/fundest/.

Lowe, David G. (Nov. 2004). "Distinctive Image Features from Scale-Invariant Keypoints." In: *International Journal of Computer Vision* 60.2, pp. 91–110.

Lucas, Bruce D. and Takeo Kanade (Apr. 1981). "An Iterative Image Registration Technique with an Application to Stereo Vision." In: *Proceedings of the 1981 DARPA Image Understanding Workshop*, pp. 121–130.

Luo, Chenxu, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille (2019). "Every Pixel Counts++: Joint Learning of Geometry and Motion with 3D Holistic Understanding." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Lv, Zhaoyang, Kihwan Kim, Alejandro Troccoli, Deqing Sun, James M. Rehg, and Jan Kautz (2018). "Learning Rigidity in Dynamic Scenes with a Moving Camera for 3D Motion Field Estimation." In: *Proceedings of the 15th European Conference*

on Computer Vision. Ed. by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss. Lecture Notes in Computer Science. Springer, pp. 468–484.

Ma, Wei-Chiu, Shenlong Wang, Rui Hu, Yuwen Xiong, and Raquel Urtasun (June 2019). "Deep Rigid Instance Scene Flow." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Long Beach, California, pp. 3614–3622.

Martinez-Cantin, Ruben (Jan. 2014). "BayesOpt: A Bayesian Optimization Library for Nonlinear Optimization, Experimental Design and Bandits." In: *Journal of Machine Learning Research* 15, pp. 3735–3739.

Maurer, Daniel and Andrés Bruhn (2018). "ProFlow: Learning to Predict Optical Flow." In: *Proceedings of the British Machine Vision Conference*.

Maurer, Daniel, Nico Marniok, Bastian Goldluecke, and Andrés Bruhn (2018). "Structure-from-Motion-Aware PatchMatch for Adaptive Optical Flow Estimation." In: *Proceedings of the 15th European Conference on Computer Vision*. Vol. 11212. Lecture Notes in Computer Science. Springer, pp. 575–592.

Mayer, Nikolaus, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox (June 2016). "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, Nevada, pp. 4040–4048.

Meister, Simon, Junhwa Hur, and Stefan Roth (Feb. 2018). "UnFlow: Unsupervised Learning of Optical Flow with a Bidirectional Census Loss." In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. New Orleans, Louisiana, pp. 7251–7259.

Mémin, Etienne and Patrick Pérez (1998). "Dense Estimation and Object-Based Segmentation of the Optical Flow with Robust Techniques." In: *IEEE Transactions on Image Processing* 7.5, pp. 703–719.

Menze, Moritz and Andreas Geiger (June 2015). "Object Scene Flow for Autonomous Vehicles." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Boston, Massachusetts, pp. 3061–3070.

Menze, Moritz, Christian Heipke, and Andreas Geiger (2015a). "Discrete Optimization for Optical Flow." In: *Proceedings of the 37th German Conference on Pattern Recognition*. Lecture Notes in Computer Science. Springer, pp. 16–28.

– (2015b). "Joint 3D Estimation of Vehicles and Scene Flow." In: *ISPRS Workshop on Image Sequence Analysis (ISA)*.

– (2018). "Object Scene Flow." In: *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)* 140, pp. 60–76.

Miksik, Ondrej, Daniel Munoz, J. Andrew Bagnell, and Martial Hebert (2013). "Efficient Temporal Consistency for Streaming Video Scene Analysis." In: *Proceedings of the IEEE International Conference on Robotics and Automation*.

Mittal, Himangi, Brian Okorn, and David Held (June 2020). "Just Go with the Flow: Self-Supervised Scene Flow Estimation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Virtual, pp. 11177–11185.

Mohamed, Mahmoud A., M. Hossein Mirabdollah, and Bärbel Mertsching (2015). "Differential Optical Flow Estimation Under Monocular Epipolar Line Constraint." In: *Proceedings of the International Conference on Computer Vision Systems*.

Neoral, Michal, Jan Šochman, and Jiří Matas (2018). "Continual Occlusions and Optical Flow Estimation." In: *Proceedings of the Fourteenth Asian Conference on Computer Vision*.

Neuhold, Gerhard, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder (Oct. 2017). "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes." In: *Proceedings of the Sixteenth IEEE International Conference on Computer Vision*. 2017, pp. 4990–4999.

Nilsson, David and Cristian Sminchisescu (June 2018). "Semantic Video Segmentation by Gated Recurrent Flow Propagation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, Utah, pp. 6819–6828.

Nowozin, Sebastian, Peter V. Gehler, and Christoph H. Lampert (2010). "On Parameter Learning in CRF-Based Approaches to Object Class Image Segmentation." In: *Proceedings of the 11th European Conference on Computer Vision*. Vol. 6316. Lecture Notes in Computer Science. Springer, pp. 98–111.

Papenberg, Nils, Andrés Bruhn, Thomas Brox, Stephan Didas, and Joachim Weickert (Apr. 2006). "Highly Accurate Optic Flow Computation with Theoretically Justified Warping." In: *International Journal of Computer Vision* 67.2, pp. 141–158.

Paul, Matthieu, Christoph Mayer, Luc Van Gool, and Radu Timofte (Mar. 2020). "Efficient Video Semantic Segmentation with Labels Propagation and Refinement." In: *IEEE Winter Conference on Applications of Computer Vision*. Aspen, CO, pp. 2873–2882.

Pérez-Rúa, Juan-Manuel, Tomas Crivelli, Patrick Bouthemy, and Patrick Pérez (June 2016). "Determining Occlusions from Space and Time Image Reconstructions." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, Nevada, pp. 1382–1391.

Qiao, Yi-Ling, Lin Gao, Yukun Lai, Fang-Lue Zhang, Ming-Ze Yuan, and Shihong Xia (Sept. 2018). "SF-Net: Learning Scene Flow from RGB-D Images with CNNs." In: *Proceedings of the British Machine Vision Conference*. Newcastle upon Tyne, UK.

Quiroga, Julian, Thomas Brox, Frédéric Devernay, and James Crowley (2014). "Dense Semi-rigid Scene Flow Estimation from RGBD Images." In: *Proceedings of the 13th European Conference on Computer Vision*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Lecture Notes in Computer Science. Springer, pp. 567–582.

Ranftl, René, Kristian Bredies, and Thomas Pock (2014). "Non-Local Total Generalized Variation for Optical Flow Estimation." In: *Proceedings of the 13th European Conference on Computer Vision*. Vol. 8689. Lecture Notes in Computer Science. Springer, pp. 439–454.

Ranjan, Anurag and Michael J. Black (July 2017). "Optical Flow Estimation Using a Spatial Pyramid Network." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii.

Ranjan, Anurag, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J. Black (June 2019). "Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Long Beach, California, pp. 12240–12249.

Ren, Zhe, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha (2017a). "Unsupervised Deep Learning for Optical Flow Estimation." In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 1495–1501.

Ren, Zhile, Orazio Gallo, Deqing Sun, Ming-Hsuan Yang, Erik B Sudderth, and Jan Kautz (Mar. 2019). "A Fusion Approach for Multi-Frame Optical Flow Estimation." In: *IEEE Winter Conference on Applications of Computer Vision*. Waikoloa Village, HI, pp. 2077–2086.

Ren, Zhile, Deqing Sun, Jan Kautz, and Erik Sudderth (2017b). "Cascaded Scene Flow Prediction Using Semantic Segmentation." In: *Proceedings of the International Conference on 3D Vision*, pp. 225–233.

Revaud, Jérôme, Philippe Weinzaepfel, Zaïd Harchaoui, and Cordelia Schmid (Dec. 2015). "EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow." In: *Proceedings of the Fifteenth IEEE International Conference on Computer Vision*. Santiago, Chile, pp. 1164–1172.

Richter, Stephan R., Zeeshan Hayder, and Vladlen Koltun (Oct. 2017). "Playing for Benchmarks." In: *Proceedings of the Sixteenth IEEE International Conference on Computer Vision*. Venice, Italy, pp. 2213–2222.

Richter, Stephan R., Vibhav Vineet, Stefan Roth, and Vladlen Koltun (2016). "Playing for Data: Ground Truth from Computer Games." In: *Proceedings of the 14th European Conference on Computer Vision*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Vol. 9906. Lecture Notes in Computer Science. Springer, pp. 102–118.

Rogers, Brian and Maureen Graham (1979). "Motion Parallax as an Independent Cue for Depth Perception." In: *Perception* 9.2, pp. 125–134.

Ros, German, Sebastian Ramos, Manuel Granados, Amir Bakhtiary, David Vazquez, and Antonio M. Lopez (2015). "Vision-Based Offline-Online Perception Paradigm for Autonomous Driving." In: *IEEE Winter Conference on Applications of Computer Vision*.

Rother, Carsten, Vladimir Kolmogorov, Victor Lempitsky, and Martin Szummer (June 2007). "Optimizing Binary MRFs via Extended Roof Duality." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Minneapolis, Minnesota.

Saxena, Rohan, René Schuster, Oliver Wasenmüller, and Didier Stricker (June 2019). "PWOC-3D: Deep Occlusion-Aware End-to-End Scene Flow Estimation." In: *Proceedings of the IEEE Intelligent Vehicles Symposium*. Paris, France, pp. 324–331.

Scharwächter, Timo, Markus Enzweiler, Uwe Franke, and Stefan Roth (2014). "Stixmantics: A Medium-Level Model for Real-Time Semantic Scene Understanding." In: *Proceedings of the 13th European Conference on Computer Vision*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Vol. 8693. Lecture Notes in Computer Science. Springer, pp. 533–548.

Schuster, René, Christian Bailer, Oliver Wasenmüller, and Didier Stricker (2018a). "FlowFields++: Accurate Optical Flow Correspondences Meet Robust Interpolation." In: *Proceedings of the IEEE International Conference on Image Processing*, pp. 1463–1467.

Schuster, René, Oliver Wasenmüller, Georg Kuschk, Christian Bailer, and Didier Stricker (Mar. 2018b). "SceneFlowFields: Dense Interpolation of Sparse Scene Flow

Correspondences." In: *IEEE Winter Conference on Applications of Computer Vision*. Lake Tahoe, NV/CA, pp. 1056–1065.

Sengupta, Sunando, Eric Greveson, Ali Shahrokni, and Philip H. S. Torr (2013). "Urban 3D Semantic Modelling Using Stereo Vision." In: *Proceedings of the IEEE International Conference on Robotics and Automation*.

Sevilla-Lara, Laura, Deqing Sun, Varun Jampani, and Michael J. Black (June 2016). "Optical Flow with Semantic Segmentation and Localized Layers." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, Nevada, pp. 3889–3898.

Shu, Chang, Kun Yu, Zhixiang Duan, and Kuiyuan Yang (2020). "Feature-Metric Loss for Self-Supervised Learning of Depth and Egomotion." In: *Proceedings of the 16th European Conference on Computer Vision*. Springer, pp. 572–588.

Song, Shuran and Jianxiong Xiao (2014). "Sliding Shapes for 3D Object Detection in Depth Images." In: *Proceedings of the 13th European Conference on Computer Vision*. Vol. 8694. Lecture Notes in Computer Science. Springer, pp. 634–651.

Stein, Fridtjof (2004). "Efficient Computation of Optical Flow Using the Census Transform." In: *Pattern Recognition, Proceedings of the 26th DAGM-Symposium*. Ed. by C. Rasmussen, H. Bülthoff, B. Schölkopf, and M. Giese. Vol. 3175. Lecture Notes in Computer Science. Springer, pp. 79–86.

Strecha, Christoph, Rik Fransens, and Luc Van Gool (2004). "A Probabilistic Approach to Large Displacement Optical Flow and Occlusion Detection." In: *Statistical Methods in Video Processing*, pp. 71–82.

Sturgess, Paul, Karteek Alahari, Lubor Ladicky, and Philip H. S. Torr (Sept. 2012). "Combining Appearance and Structure from Motion Features for Road Scene Understanding." In: *Proceedings of the British Machine Vision Conference*. Surrey, UK.

Sun, Deqing, Ce Liu, and Hanspeter Pfister (June 2014a). "Local Layering for Joint Motion Estimation and Occlusion Detection." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, Ohio, pp. 1098–1105.

Sun, Deqing, Stefan Roth, and Michael J. Black (June 2010a). "Secrets of Optical Flow Estimation and Their Principles." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. San Francisco, California, pp. 2432–2439.

– (Jan. 2014b). "A Quantitative Analysis of Current Practices in Optical Flow Estimation and the Principles Behind Them." In: *International Journal of Computer Vision* 106.2, pp. 115–137.

Sun, Deqing, Erik B. Sudderth, and Michael J. Black (2010b). "Layered Image Motion with Explicit Occlusions, Temporal Consistency, and Depth Ordering." In: *Advances in Neural Information Processing Systems*. Ed. by J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta. Vol. 23, pp. 2226–2234.

Sun, Deqing, Daniel Vlasic, Charles Herrmann, Varun Jampani, Michael Krainin, Huiwen Chang, Ramin Zabih, William T. Freeman, and Ce Liu (June 2021). "AutoFlow: Learning a better training set for optical flow." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Virtual, pp. 10093–10102.

Sun, Deqing, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz (June 2018). "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." In: *Proceedings*

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, Utah, pp. 8934–8943.

– (June 2020). "Models Matter, So Does Training: An Empirical Study of CNNs for Optical Flow Estimation." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.6, pp. 1408–1423.

Taniai, Tatsunori, Yasuyuki Matsushita, and Takeshi Naemura (June 2014). "Graph Cut Based Continuous Stereo Matching Using Locally Shared Labels." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, Ohio, pp. 1613–1620.

Taniai, Tatsunori, Yasuyuki Matsushita, Yoichi Sato, and Takeshi Naemura (Oct. 2017). "Continuous 3D Label Stereo Matching Using Local Expansion Moves." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.11, pp. 2725–2739.

Teed, Zachary and Jia Deng (2020). "RAFT: Recurrent All-Pairs Field Transforms for Optical Flow." In: *Proceedings of the 16th European Conference on Computer Vision*. Springer.

Thakur, Ravi Kumar and Snehasis Mukherjee (2018). "SceneEDNet: A Deep Learning Approach for Scene Flow Estimation." In: *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 394–399.

Tompson, Jonathan J., Arjun Jain, Yann LeCun, and Christoph Bregler (2014). "Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation." In: *Advances in Neural Information Processing Systems*. Vol. 27, pp. 1799–1807.

Tonioni, Alessio, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano (Oct. 2019). "Unsupervised domain adaptation for depth prediction from images." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.10, pp. 2396–2409.

Tosi, Fabio, Filippo Aleotti, Pierluigi Zama Ramirez, Matteo Poggi, Samuele Salti, Luigi Di Stefano, and Stefano Mattoccia (June 2020). "Distilled Semantics for Comprehensive Scene Understanding from Videos." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Virtual, pp. 4654–4665.

Tu, Zhigang, Wei Xie, Dejun Zhang, Ronald Poppe, Remco C. Veltkamp, Baoxin Li, and Junsong Yuan (Mar. 2019). "A Survey of Variational and CNN-Based Optical Flow Techniques." In: *Signal Processing: Image Communication* 72, pp. 9–24.

Ummenhofer, Benjamin, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox (July 2017). "DeMoN: Depth and Motion Network for Learning Monocular Stereo." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, pp. 5622–5631.

Unger, Markus, Manuel Werlberger, Thomas Pock, and Horst Bischof (June 2012). "Joint Motion Estimation and Segmentation of Complex Scenes with Label Costs and Occlusion Modeling." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Providence, Rhode Island, pp. 1878–1885.

Vedula, Sundar, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade (Sept. 1999). "Three-Dimensional Scene Flow." In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Kerkyra, Greece, pp. 722–729.

– (Mar. 2005). "Three-Dimensional Scene Flow." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.3, pp. 475–480.

Vogel, Christoph, Stefan Roth, and Konrad Schindler (2014). "View-Consistent 3D Scene Flow Estimation over Multiple Frames." In: *Proceedings of the 13th European Conference on Computer Vision*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Vol. 8692. Lecture Notes in Computer Science. Springer, pp. 263–278.

Vogel, Christoph, Konrad Schindler, and Stefan Roth (2013a). "An Evaluation of Data Costs for Optical Flow." In: *Proceedings of the 35th German Conference on Pattern Recognition*. Ed. by J. Weickert, M. Hein, and B. Schiele. Vol. 8142. Lecture Notes in Computer Science. Springer, pp. 343–353.

– (Dec. 2013b). "Piecewise Rigid Scene Flow." In: *Proceedings of the Fourteenth IEEE International Conference on Computer Vision*. Sydney, Australia, pp. 1377–1384.

– (Oct. 2015). "3D Scene Flow Estimation with a Piecewise Rigid Scene Model." In: *International Journal of Computer Vision* 115.1, pp. 1–28.

Wang, Shenlong, Sean Ryan Fanello, Christoph Rhemann, Shahram Izadi, and Pushmeet Kohli (June 2016). "The Global Patch Collider." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, Nevada, pp. 127–135.

Wang, Xiaolong, Allan Jabri, and Alexei A. Efros (June 2019a). "Learning correspondence from the cycle-consistency of time." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Long Beach, California, pp. 2566–2576.

Wang, Yang, Peng Wang, Zhenheng Yang, Chenxu Luo, Yi Yang, and Wei Xu (June 2019b). "UnOS: Unified Unsupervised Optical-Flow and Stereo-Depth Estimation by Watching Videos." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Long Beach, California, pp. 8071–8081.

Wang, Yang, Yi Yang, Zhenheng Yang, Liang Zhao, and Wei Xu (June 2018). "Occlusion Aware Unsupervised Learning of Optical Flow." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4884–4893.

Wang, Zhou, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli (Apr. 2004). "Image Quality Assessment: From Error Visibility to Structural Similarity." In: *IEEE Transactions on Image Processing* 13.4, pp. 600–612.

Wang, Zirui, Shuda Li, Henry Howard-Jenkins, Victor Prisacariu, and Min Chen (Mar. 2020). "FlowNet3D++: Geometric Losses for Deep Scene Flow Estimation." In: *IEEE Winter Conference on Applications of Computer Vision*. Aspen, CO, pp. 91–98.

Wannenwetsch, Anne S. and Stefan Roth (June 2020). "Probabilistic Pixel-Adaptive Refinement Networks." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. to appear. Virtual.

Wedel, Andreas, Thomas Brox, Tobi Vaudrey, Clemens Rabe, Uwe Franke, and Daniel Cremers (Oct. 2011). "Stereoscopic Scene Flow Computation for 3D Motion Understanding." In: *International Journal of Computer Vision* 95.1, pp. 29–51.

Wedel, Andreas, Clemens Rabe, Tobi Vaudrey, Thomas Brox, Uwe Franke, and Daniel Cremers (2008). "Efficient Dense Scene Flow from Sparse or Dense Stereo Data." In: *Proceedings of the Tenth European Conference on Computer Vision*. Ed. by D. Forsyth, P. Torr, and A. Zisserman. Lecture Notes in Computer Science. Springer, pp. 739–751.

Weickert, Joachim and Christoph Schnörr (Dec. 2001). "A Theoretical Framework for Convex Regularizers in PDE-Based Computation of Image Motion." In: *International Journal of Computer Vision* 45.3, pp. 245–264.

Werlberger, Manuel, Thomas Pock, and Horst Bischof (June 2010). "Motion Estimation with Non-Local Total Variation Regularization." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. San Francisco, California.

Werlberger, Manuel, Werner Trobin, Thomas Pock, Andreas Wedel, Daniel Cremers, and Horst Bischof (Sept. 2009). "Anisotropic Huber-L1 Optical Flow." In: *Proceedings of the British Machine Vision Conference*. London, UK.

Woodford, Oliver J., Philip H. S. Torr, Ian D. Reid, and Andrew W. Fitzgibbon (June 2008). "Global Stereo Reconstruction under Second Order Smoothness Priors." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Anchorage, Alaska.

Wu, Wenxuan, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin (2020). "PointPWC-Net: Cost Volume on Point Clouds for (Self-)Supervised Scene Flow Estimation." In: *Proceedings of the 16th European Conference on Computer Vision*. Springer.

Wulff, Jonas, Laura Sevilla-Lara, and Michael J. Black (July 2017). "Optical Flow in Mostly Rigid Scenes." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, pp. 6911–6920.

Xia, Wei, Csaba Domokos, Jian Dong, Loong-Fah Cheong, and Shuicheng Yan (Dec. 2013). "Semantic Segmentation without Annotating Segments." In: *Proceedings of the Fourteenth IEEE International Conference on Computer Vision*. Sydney, Australia, pp. 2176–2183.

Xiao, Degui, Qiuwei Yang, Bing Yang, and Wei Wei (2017). "Monocular Scene Flow Estimation via Variational Method." In: *Multimedia Tools and Applications* 76.8, pp. 10575–10597.

Xiao, Jiangjian, Hui Cheng, Harpreet S. Sawhney, Cen Rao, and Michael A. Isnardi (2006). "Bilateral Filtering-Based Optical Flow Estimation with Occlusion Detection." In: *Proceedings of the Ninth European Conference on Computer Vision*. Ed. by A. Leonardis, H. Bischof, and A. Pinz. Vol. 3951. Lecture Notes in Computer Science. Springer, pp. 211–224.

Xiao, Taihong, Jinwei Yuan, Deqing Sun, Qifei Wang, Xin-Yu Zhang, Kehan Xu, and Ming-Hsuan Yang (2020). "Learnable Cost Volume Using the Cayley Representation." In: *Proceedings of the 16th European Conference on Computer Vision*. Springer.

Xiong, Yuwen, Mengye Ren, Wenyuan Zeng, and Raquel Urtasun (Oct. 2021). "Self-supervised representation learning from flow equivariance." In: *Proceedings of the Seventeenth IEEE International Conference on Computer Vision*. Virtual, pp. 10191–10200.

Xu, Jia, René Ranftl, and Vladlen Koltun (July 2017). "Accurate Optical Flow via Direct Cost Volume Processing." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, pp. 5807–5815.

Xu, Li, Jiaya Jia, and Yasuyuki Matsushita (2011). "Motion Detail Preserving Optical Flow Estimation." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.9, pp. 1744–1757.

Yamaguchi, Koichiro, David McAllester, and Raquel Urtasun (June 2013). "Robust Monocular Epipolar Flow Estimation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Portland, Oregon, pp. 1862–1869.

Yamaguchi, Koichiro, David McAllester, and Raquel Urtasun (2014). "Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation." In: *Proceedings of the 13th European Conference on Computer Vision*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Vol. 8693. Lecture Notes in Computer Science. Springer, pp. 756–771.

Yan, Zike and Xuezhi Xiang (2016). "Scene Flow Estimation: A survey." In: *arXiv:1612.02590 [cs.CV]*.

Yang, Gengshan and Deva Ramanan (2019). "Volumetric Correspondence Networks for Optical Flow." In: *Advances in Neural Information Processing Systems*, pp. 793–803.

– (June 2020). "Upgrading Optical Flow to 3D Scene Flow Through Optical Expansion." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Virtual, pp. 1331–1340.

Yang, Jiaolong and Hongdong Li (June 2015). "Dense, Accurate Optical Flow Estimation with Piecewise Parametric Model." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Boston, Massachusetts, pp. 1019–1027.

Yang, Yanchao and Stefano Soatto (July 2017). "S2F: Slow-to-fast Interpolator Flow." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, pp. 3767–3776.

Yang, Zhenheng, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia (2018). "Every Pixel Counts: Unsupervised Geometry Learning with Holistic 3D Motion Understanding." In: *ECCV Workshops*. Ed. by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss. Lecture Notes in Computer Science. Springer, pp. 691–709.

Yao, Jian, Marko Boben, Sanja Fidler, and Raquel Urtasun (June 2015). "Real-Time Coarse-to-Fine Topologically Preserving Segmentation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Boston, Massachusetts, pp. 2947–2955.

Yin, Zhichao, Trevor Darrell, and Fisher Yu (June 2019). "Hierarchical Discrete Distribution Decomposition for Match Density Estimation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Long Beach, California, pp. 6044–6053.

Yin, Zhichao and Jianping Shi (June 2018). "GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, Utah, pp. 1983–1992.

Yu, Fisher and Vladlen Koltun (May 2016). "Multi-Scale Context Aggregation by Dilated Convolutions." In: *Proceedings of the International Conference on Learning Representations*. San Juan, Puerto Rico.

Yu, Jason J., Adam W. Harley, and Konstantinos G. Derpanis (2016). "Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constancy and Motion Smoothness." In: *ECCV Workshops*. Vol. 9907. Lecture Notes in Computer Science. Springer, pp. 3–10.

Zabih, Ramin and John Woodfill (1994). "Non-parametric Local Transforms for Computing Visual Correspondence." In: *Proceedings of the Third European Conference on Computer Vision*. Ed. by J.-O. Eklundh. Vol. 801. Lecture Notes in Computer Science. Springer, pp. 151–158.

Zach, Christopher, Thomas Pock, and Horst Bischof (2007). "A Duality Based Approach for Realtime TV-L1 Optical Flow." In: *Pattern Recognition, Proceedings of the 29th DAGM-Symposium*. Ed. by Fred A. Hamprecht, Christoph Schnörr, and Bernd Jähne. Vol. 4713. Lecture Notes in Computer Science. Springer, pp. 214–223.

Zhang, Chenxi, Liang Wang, and Ruigang Yang (2010). "Semantic Segmentation of Urban Scenes Using Dense Depth Maps." In: *Proceedings of the 11th European Conference on Computer Vision*. Vol. 6314. Lecture Notes in Computer Science. Springer, pp. 708–721.

Zhang, Feihu, Oliver J. Woodford, Victor Adrian Prisacariu, and Philip H.S. Torr (Oct. 2021). "Separable flow: Learning motion cost volumes for optical flow estimation." In: *Proceedings of the Seventeenth IEEE International Conference on Computer Vision*. Virtual, pp. 10807–10817.

Zhang, Ye and Chandra Kambhamettu (Dec. 2001). "On 3D Scene Flow and Structure Estimation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Kauai, Hawaii, pp. 3526–3533.

Zhao, Hengshuang, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia (July 2017). "Pyramid Scene Parsing Network." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, pp. 2881–2890.

Zhao, Shengyu, Yilun Sheng, Yue Dong, Eric I Chang, and Yan Xu (June 2020). "MaskFlownet: Asymmetric Feature Matching with Learnable Occlusion Mask." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Virtual, pp. 6278–6287.

Zhou, Huizhong, Benjamin Ummenhofer, and Thomas Brox (2018). "DeepTAM: Deep Tracking and Mapping." In: *Proceedings of the 15th European Conference on Computer Vision*. Vol. 11220. Lecture Notes in Computer Science. Springer, pp. 851–868.

Zhou, Tinghui, Matthew Brown, Noah Snavely, and David G. Lowe (July 2017). "Unsupervised Learning of Depth and Ego-Motion from Video." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, pp. 1851–1858.

Zhu, Alex Zihao, Wenxin Liu, Ziyun Wang, Vijay Kumar, and Kostas Daniilidis (June 2019). "Robustness Meets Deep Learning: An End-to-End Hybrid Pipeline for Unsupervised Learning of Egomotion." In: *CVPR Workshops*. Long Beach, California.

Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros (Oct. 2017a). "Unpaired image-to-image translation using cycle-consistent adversarial networks." In: *Proceedings of the Sixteenth IEEE International Conference on Computer Vision*. Venice, Italy, pp. 2223–2232.

Zhu, Xizhou, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei (July 2017b). "Deep Feature Flow for Video Recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, pp. 4141–4150.

Zhu, Yi, Zhenzhong Lan, Shawn Newsam, and Alexander G. Hauptmann (2017c). "Guided Optical Flow Learning." In: *CVPR 2017 Workshops*.

Zhu, Yi and Shawn D. Newsam (2017). "DenseNet for Dense Flow." In: *Proceedings of the IEEE International Conference on Image Processing*, pp. 790–794.

Zou, Yang, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang (2018a). "Unsupervised Domain Adaptation for Semantic Segmentation via Class-Balanced Self-Training." In: *Proceedings of the 15th European Conference on Computer Vision*. Springer, pp. 289–305.

Zou, Yuliang, Zelun Luo, and Jia-Bin Huang (2018b). "DF-Net: Unsupervised Joint Learning of Depth and Flow Using Cross-Task Consistency." In: *Proceedings of the 15th European Conference on Computer Vision*. Ed. by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss. Lecture Notes in Computer Science. Springer, pp. 36–53.

Junhwa Hur

INFO. junhwa.hur@gmail.com / Google Scholar / GitHub / Portfolio Webpage

RESEARCH INTEREST
3D Dynamic Scene Understanding: Semantic segmentation, Motion, Depth, 3D reconstruction
Learning with Limited Supervision: Self-supervised learning, Semi-supervised learning

PROFESSIONAL EXPERIENCE

42dot, *Seoul, Korea* — Oct. 2021 – Jun. 2022
Research Internship at Autonomous Intelligence
- Working on a surround-view 3D reconstruction project

Technische Universität Darmstadt, *Darmstadt, Germany* — Oct. 2015 – Oct. 2020
Doctoral Research Assistant (Supervised by Prof. Stefan Roth, Ph.D.)
- Researched multi-task learning for 3D dynamic scene understanding: motion, depth, occlusion, and semantic segmentation using (self-)supervised learning

Korea Institute of Science and Technology (KIST), *Seoul, South Korea* — Feb. 2014 – Aug. 2015
Internship at Imaging Media Research Center
- Developed a pipeline for RGB-D-based 3D deformable object modeling (correspondence and pose estimation, mesh generation, and loop closure).

Seoul National University, *Seoul, South Korea* — Sep. 2011 – Dec. 2013
Research Assistant at Vehicle Intelligence Lab
- Researched computer vision algorithms for autonomous driving and deployed them on self-driving cars.

EDUCATION

Technische Universität Darmstadt, *Darmstadt, Germany* — 2015 – 2022
Ph.D. candidate in Computer Science
- Dissertation: Joint Motion, Semantic Segmentation, Occlusion, and Depth Estimation

Seoul National University, *Seoul, South Korea* — 2011 – 2013
M.Sc. in Electrical and Computer Engineering
- Thesis: Multi-Lane Detection in Highway and Urban Driving Environment

Pohang University of Science and Technology, *Pohang, South Korea* — 2007 – 2011
B.Sc. in Electronics and Electrical Engineering, *Magna Cum Laude*

PUBLICATIONS (HYPERLINKED)

Junhwa Hur and Stefan Roth, "Self-Supervised Multi-Frame Monocular Scene Flow", **CVPR**, 2021

Junhwa Hur and Stefan Roth, "Self-Supervised Monocular Scene Flow Estimation", **CVPR**, 2020, **Oral Presentation**

Junhwa Hur and Stefan Roth, "Optical Flow Estimation in the Deep Learning Age", as a book chapter in Modelling Human Motion, Springer, 2020

Junhwa Hur and Stefan Roth, "Iterative Residual Refinement for Joint Optical Flow and Occlusion Estimation", **CVPR**, 2019

Simon Meister, **Junhwa Hur** and Stefan Roth, "UnFlow: Unsupervised Learning of Optical Flow with a Bidirectional Census Loss", **AAAI**, 2018, **Oral Presentation**

Junhwa Hur and Stefan Roth, "MirrorFlow: Exploiting Symmetries in Joint Optical Flow and Occlusion Estimation", **ICCV**, 2017

Junhwa Hur and Stefan Roth, "Joint Optical Flow and Temporally Consistent Semantic Segmentation", **ECCV Workshop** on CVRSUAD, 2016, **Best paper award**

Junhwa Hur, Hwasup Lim, Changsoo Park, Sang Chul Ahn, "Generalized Deformable Spatial Pyramid: Geometry-Preserving Dense Correspondence Estimation", **CVPR**, 2015

Junhwa Hur, Hwasup Lim, Sang Chul Ahn, "3D Deformable Spatial Pyramid for Dense 3D Motion Flow of Deformable Object", **ISVC**, 2014

Seung-Nam Kang, Soo-Mok Lee, **Junhwa Hur**, and Seung-Woo Seo, "Multi-lane Detection based on Accurate Geometric Lane Estimation in Highway Scenarios", **IV**, 2014

Junhwa Hur, Seung-Nam Kang, and Seung-Woo Seo, "Multi-lane Detection in Urban Driving Environments using Conditional Random Fields", **IV**, 2013.

Junhwa Hur, "Multi-lane Detection in Highway and Urban Driving Environment", Master's thesis, Seoul National University, 2013

Teaching Experience

Teaching Assistantship, *TU Darmstadt, Germany* 2015 – 2020

- Computer Vision I & II
- Advanced Topics in Computer Vision Machine Learning
- Project Lab Deep Learning for Computer Vision – supervised 4 team projects (Self-supervised learning, Semantic image inpainting using GAN, Monocular depth, Optical flow)
- B.Sc. & M.Sc. Thesis Supervision – supervised 5 students (Scene flow, Monocular depth, Dataset bias analysis, Moving object detection, Multi-task learning)

Awards and Honors

Outstanding Reviewer Award: CVPR (2018, 2019, 2020, 2022), ICCV (2021), ECCV (2020), ACCV (2020)
Doctoral Consortium, CVPR 2020
Best Paper Award, 21. Darmstädter Computer Graphik Abend 2019, Impact on Science
Best Paper Award, 20. Darmstädter Computer Graphik Abend 2018, Impact on Science
Best Paper Award, ECCV Workshops 2016 - Computer Vision for Road Scene Understanding and Autonomous Driving
2nd Place Prize, Korea Autonomous Vehicle Contest 2013
National Science and Engineering Scholarship (covering full tuitions), KFAS, 2007 – 2011
Merit-based Scholarship, POSTECH, 2007 – 2008

Reviewer Activity

Conference: ICLR, NeurIPS, CVPR, ICCV, ECCV, ACCV, WACV, ICRA, IROS
Journal: T-PAMI, T-IP, RA-L, PR, T-CSVT

Skill

C/C++, Python, Matlab, PyTorch, TensorFlow

Language

Korean (Native, Citizenship), English (Fluent), German (Intermediate, Permanent residency)