CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

F3

Faculty of Electrical Engineering
Department of Cybernetics

Master Thesis

Self-Supervised Optical Flow Learning

Bc. Tomáš Novák

Supervisor: prof. Ing. Jiří Matas, Ph.D.
Field of study: Open Informatics
Subfield: Computer Vision and Image Processing
January 2020

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Novák Tomáš**	Personal ID number:	**434867**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Cybernetics**		
Study program:	**Open Informatics**		
Branch of study:	**Computer Vision and Image Processing**		

II. Master’s thesis details

Master’s thesis title in English:

Self-Supervised Optical Flow Learning

Master’s thesis title in Czech:

Učení optického toku bez učitele

Guidelines:

The goal of this thesis is to design, implement and evaluate a method for (partly or fully) self-supervised learning of artificial neural network for optical flow estimation.
Workplan:
1. Get familiar with the state-of-the-art in self-supervised optical flow training methods.
2. Get familiar with a recent well-performing optic flow method, e.g. PWC-Net [1].
3. Propose and implement a method for (partly or fully) self-supervised learning of optical flow network.
4. Evaluate the proposed method on some suitable dataset(s).
5. Discuss the results, analyzing strengths, weaknesses and failure modes of the proposed method.

Bibliography / sources:

[1] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, “PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8934–8943.
[2] P. Liu, M. Lyu, I. King, and J. Xu, “SelFlow: Self-Supervised Learning of Optical Flow,” arXiv:1904.09117 [cs], Apr. 2019.
[3] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu, “Occlusion Aware Unsupervised Learning of Optical Flow,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4884–4893.
[4] S. Meister, J. Hur, and S. Roth, “UnFlow: Unsupervised Learning of Optical Flow With a Bidirectional Census Loss,” in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[5] J. J. Yu, A. W. Harley, and K. G. Derpanis, “Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constancy and Motion Smoothness,” in Computer Vision – ECCV 2016 Workshops, 2016, pp. 3–10.

Name and workplace of master’s thesis supervisor:

prof. Ing. Jiří Matas, Ph.D., Visual Recognition Group, FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **24.05.2019** Deadline for master's thesis submission: **07.01.2020**

Assignment valid until: **19.02.2021**

prof. Ing. Jiří Matas, Ph.D. Supervisor’s signature	doc. Ing. Tomáš Svoboda, Ph.D. Head of department’s signature	prof. Ing. Pavel Ripka, CSc. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

________.__________________
Date of assignment receipt

____
Student’s signature

Acknowledgements

Firstly, I would like to thank the supervisor of this work Jiří Matas for his support during the project. I also thank Jan Šochman for his great help and guidance throughout the whole work. Furthermore, my thanks go to Michal Neoral and Jonáš Šerých who were always ready to answer my questions.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 5. January 2020

Abstract

Convolutional neural networks currently dominate in optical flow estimation. Neural network learning methods are categorized to three groups: *supervised* needing inputs and desired outputs, *self-supervised* needing just inputs and *semi-supervised* attempting to combine both.

This work proposes a new method of semi-supervised optical flow learning. The method formulates the training optimization as constrained gradient descent on a supervised loss function that includes self-supervised terms. In the self-supervised domain, a systematic study of selected current practices is done. Specifically, three photometric difference measures are tested - brightness difference, census transform and structural similarity. Current research suggests that occlusion handling plays a role for self-supervised learning. Two methods are tested - forward backward consistency occlusion detection from UnFlow [32] and three-frame occlusion reasoning from Janai et al. [31]. Apart from these techniques, we also test the training dataset size effect and forward-backward consistency loss function term [32].

The experiments regarding semi-supervision show that including the unsupervised objective with the proposed method significantly improves the estimation on a distant domain while maintaining the performance on the original domain. More specifically, the error decrease is demonstrated on an artistic-like Creative Flow+ dataset [42] while the model maintain its accuracy on the popular Sintel dataset [11]. Surprisingly, the effect is observed even wihtout using any Creative Flow+ samples.

The self-supervised training experiments show that learning with census photometric difference leads to better accuracy on all tested datasets. Out of the two occlusion handling methods, none significantly increases the performance. The results suggest that the methods are unable to accurately detect occlusions. The experiments show that a large amount of training data does not necessarily lead to a performance increase. Surprisingly, training on as little as eighty frame pairs does not lead to a catastrophic loss of accuracy.

Keywords: computer vision, optical flow, self-supervised training, semi-supervised training

Supervisor: prof. Ing. Jiří Matas, Ph.D.

Abstrakt

Konvoluční neuronové sítě v současnosti dominují odhadu optického toku. Metody učení neuronových sítí se dělí do tří skupin: *učení s učitelem*, které používá vstupy s požadovanými výstupy, *učení bez učitele*, které vyžaduje pouze vstupy a *kombinace učení s učitelem a bez učitele*, které se pokouší sloučit obojí.

Tato práce navrhuje novou metodu kombinace učení s učitelem a bez učitele. Metoda formuluje optimalizaci při trénování jako omezený gradientní sestup na ztrátové funkci zahrnující termy z učení bez učitele. V doméně učení bez učitele je provedena systematická studie vybraných současných technik. Konkrétně jsou testovány tři techniky měření fotometrického rozdílu - rozdíl jasu, Census transformace a structural similarity. Současný výzkum ukazuje, že zohledňování zákrytů hraje roli při učení bez učitele a proto jsou otestovány dvě metody - detekce zákrytů pomocí zpětné konzistence z UnFlow [32] a třísnímkové zohledňování okluzí z Janai et al. [31]. Kromě těchto technik testujeme také vliv velikosti trénovacího datasetu a term zpětné konzistence ve ztrátové funkci [32].

Experimenty ohledně kombinace učení s učitelem a bez učitele ukazují, že přidáním cíle z učení bez učitele pomocí navrhované metody výrazně zlepšuje odhad optického toku na vzdálené doméně a přitom zachovává přesnost na doméně výchozí. Konkrétněji je pokles chyby demonstrován na uměleckém Creative Flow+ datasetu [42], přičemž model zachovává přesnost na datasetu Sintel [11]. Překvapivě je efekt pozorován i bez použití snímků z Creative Flow+.

Experimenty s učením bez učitele ukazují, že učení s fotometrickým rozdílem určovaným pomocí Census transformace vede k větší přesnosti na všech testovaných datasetech. Ani jedna z obou testovaných metod pro zohledňování zákrytů výrazně nezvětšuje přesnost. Výsledky naznačují, že selhává schopnost přesně najít zákryty. Experimenty ukazují, že velké množství trénovacích dat nevede ve všech případech ke zlepšení přesnosti. Překvapivě, katastrofální úbytek přesnosti není zaznamenán při trénování na pouze osmdesáti párech snímků.

Klíčová slova: počítačové vidění, optický tok, učení bez učitele, kombinace učení s učitelem a bez učitele

Překlad názvu: Učení optického toku bez učitele

Contents

Figures

Tables

Chapter 1

Introduction

Supervision is the opium of the AI researcher.

Jitendra Malik, CVPR 2019

The concept of optical flow originates in psychology. It was introduced by James J. Gibson in the 1940s to describe the visual stimuli provided to animals moving through the world [43]. Later, the term was adopted by computer vision to describe a dense motion field between two consecutive frames in a video. Since the 1980s, many methods were proposed for the task of optical flow estimation from images. Horn-Schunck [1] and Lucas-Kanade [2] serve as examples of the first classical methods, but the task is still a point of intense research.

Like many other areas in computer vision, optical flow estimation is currently dominated by convolutional neural networks. Neural network training usually requires a large number of annotated samples. However, acquiring optical flow ground-truth is a non-trivial task. Fully manual annotation is extremely time-consuming and uncommon. Some datasets, e.g., KITTI [12, 22], employ sensors like Lidar to capture a 3D structure of a real-world scene and an IMU to measure ego-motion. The information is then used to determine the optical flow in frames of a calibrated camera. Nevertheless, the estimated optical flow has a relatively high level of uncertainty, optical flow on rigid moving objects requires a manual annotation (to fit a 3D model), and optical flow on non-rigid objects is virtually impossible to obtain.

Currently, the most feasible option to obtain optical flow ground-truth is to synthesize a scene using computer graphics. This is also how Sintel [11], one of the most popular optical flow benchmarks, was created. This approach provides an accurate optical flow and has the potential to create a large dataset.

A sizeable optical flow dataset *FlyingChairs*, created by synthesizing scenes with 3D models of chairs, was the critical element of the first CNN-based optical flow estimation method *FlowNet* [19]. This dataset still plays an essential role during the training of almost all supervised optical flow estimation networks. However, as all synthesized datasets, it suffers from an unknown domain shift between the modeled scenes and real-world optical flow.

Supervised learning, where neural networks are trained by being presented with examples, including ground-truth annotation, is not the only approach. Parallel to this class of learning, there are also “unsupervised” methods (also referred to as “self-supervised”[1]) that formulate the training without the need of ground-truth.

In the case of optical flow, the basic principle underpinning the unsupervised training is to formulate a loss function that evaluates the photometric consistency of the given dense correspondences. Additionally, to cope with local ambiguities and other effects, the loss function includes an optical flow smoothness term. This principle comes from the Horn-Schunck method, where a similar objective is optimized for each image pair. Some unsupervised training methods also propose to take occlusions into consideration as these corrupt the photometric consistency of optical flow - if an area is occluded in the second frame by being out-of-frame or behind another object in the scene, it is impossible to measure photometric consistency.

The main advantage of unsupervised optical flow training is that there is no need for ground-truth. This allows for training on a virtually unlimited number of data. It opens a possibility to train a wider variety of scenes with potentially improved accuracy. Furthermore, the training can be done on a specific domain, where the annotation is not available or is impossible to obtain.

These advantages motivate the research of unsupervised optical flow training methods. There are several publications working with the same main principle, but proposing different variations [31, 32, 34]. They suggest a variety of photometric consistency measures, occlusion handling methods, etc. Moreover, each is developed under slightly different conditions - diverse network architectures, training datasets, and training protocols. This prevents a direct comparison of the benefits of individual components.

The first part of this work focuses on the analysis and comparison of some of the recently proposed ideas in unsupervised optical flow training. The experiments are done under a single protocol, which allows for a comparison. Specifically, the following contributions are made regarding the understanding of unsupervised optical flow training.

- Popular photometric difference measures are compared - brightness difference, census transform, and structural similarity, and the best is selected.

[1] In the context of optical flow, both terms “unsupervised” and “self-supervised” are used to refer to the same method of training without labeled samples. This work uses both terms interchangeably.

- Two occlusion handling methods are analyzed - forward-backward consistency masking [32] and three-frame occlusion reasoning [31] and their failure-cases are analyzed.
- Proposed forward-backward consistency term [32] in unsupervised loss function is tested.
- Training dataset size influence is analyzed. A large amount of unlabeled frame pairs is collected from various sources and used as a training dataset. On the other hand, an experiment with the dataset size restricted to just 80 samples is performed.

However, evaluation on standard benchmarks like Sintel [11] or KITTI [22] shows that unsupervised training currently does note lead to an accuracy comparable with supervised training. This is most probably caused by the inability of the unsupervised loss function to fully cope with all commonly present effects like occlusions, motion blur, local ambiguity, and other.

We thus also focus on combining unsupervised training with supervised training. This strategy is called “semi-supervised” training. It has the potential to combine the accuracy of supervised training with the theoretically unlimited training dataset of unsupervised training. This is not a well-established technique in the optical flow estimation task; the experiments thus lead to valuable conclusions.

Regarding semi-supervised training, the work contributes in the following ways.

- A novel method to combine supervised and unsupervised objectives is presented. The method formulates the training as constrained gradient descent that takes gradients from loss functions of both objectives; however, skips all unsupervised samples that lead to worse performance on the supervised samples, i.e., all unsupervised gradients that have a negative dot product with the supervised gradient are omitted.
- The method is tested to combine the objectives in three scenarios - fine-tuning on a single domain, adaptation on a close domain, and distant domain adaptation.

The structure of the work is the following. First, related work is analyzed in more detail in Chapter 2, and different novel techniques are highlighted. Second, the necessary theoretical foundation for experiments is laid in Chapter 3, and the novel semi-supervision method is introduced. Chapter 4 describes the conducted experiments and lists all technical details. Lastly, Chapter 5 lists and discusses the results of the experiments.

Chapter 2

Related work

In this chapter, a closer look at major recent unsupervised/self-supervised and semi-supervised optical flow training methods is provided, and the innovative contributions of each publication are highlighted.

First, various recent unsupervised optical flow training methods are analyzed. We then focus on a special group of methods that employ the unsupervised training paradigm in a broader way and combine optical flow estimation with other tasks like depth and ego-motion estimation. Afterward, semi-supervised approaches to optical-flow training are revised.

2.1 Unsupervised optical flow methods

First, we explore all major contributions to the field of self-supervised optical flow training.

Ahmadi et Patras [23], Back2Basics [27] and DSTFlow [30] were arguably the first to introduce the idea of training an optical flow estimator network using the objective function from Horn-Schunck (H-S) method [1]. They all employ the basic brightness constancy assumption for pixel matching, with DSTFlow being the only method that also considers a gradient constancy. In all cases, the estimator is also trained to produce a smooth motion field by simple penalization of all spatial discontinuities, the same as in H-S.

However, some post-Horn-Schunck techniques can also be found. As Sun et al. [18] show, the best practice in classical methods is to use Charbonnier robust penalty for both brightness and smoothness objective to minimize the sensitivity to outliers. All mentioned methods apply this practice in the training loss function. Interestingly enough, Ahmadi et Patras [23] employ three other classical techniques - pyramidal approach with iterative refinement on each level and median filtering after each iteration.

Long et al. [24] propose to build the unsupervised estimator in a different way. They train a neural network to solve a task connected to optical flow - frame-interpolation. In a triplet of frames, the network is trained to estimate the middle frame from the first and last frames. For optical flow estimation, the network is used to compute gradients of values in the output image with respect to each input pixel. This is done in order to discover the pixels in input images that influence the pixel in the output image the most i.e., establishing correspondences.

TransFlow [28] introduces an interesting iterative approach - flow estimation is done in two stages by two separate networks. The first network estimates a homography that approximates coarse camera movements in the image. The second network then adds just the residual motion. This approach is, however, not robust - it is crafted especially for driving sequences, and it is destined to fail with general motion.

UnFlow [32] and Wang et al. [34] were the first methods to introduce occlusion reasoning into the unsupervised optical flow training process. To achieve this, both methods work with both forward and backward optical flow fields.

Wang et al. [34] build on the observation that with ideal optical flow fields, pixels that are occluded in the second image, do not adhere to any backward flow (i.e., flow from second to the first image). Based on the backward optical flow, we label as occluded those pixels in the first image that do not have a correspondence in the second image (i.e., no vector from the backward flow field points to them). Photometric term (brightness and gradient constancy in this case) is then not considered on occluded pixels in the forward flow loss function, and the presence of occluded pixels is penalized in order to avoid the *all-pixels-occluded* solution.

UnFlow [32] takes a slightly different path to determine the occluded pixels. Forward-backward optical flow consistency is considered - the basic idea is that on non-occluded pixels, one should be able to follow the forward optical flow and then get back to the same pixel with the backward flow. Since this approach seems to be popular - it is employed also in later works of Liu et al. [39, 40], we put it under analysis (Section 4.2.2). UnFlow also penalizes forward-backward inconsistencies and finds that it significantly improves results. This finding is also analyzed (Section 4.2.3).

Janai et al.[31] come with a different technique to cope with occlusions. Instead of using only the future frame to reconstruct the current frame, both neighboring frames are used (i.e., both past and future) for reconstruction. This approach assumes that all pixels in the current frame are visible at least in one of the adjacent frames, which holds in most cases. We investigate the exact influence of similar reasoning in Section 4.2.2.

DDFlow [39] is a method presenting so-called data distillation approach to training. The main goal is to train the network to estimate optical flow correctly, even on occluded pixels. The authors first train a model following the main ideas of UnFlow [32] and call it the "teacher" network. While training secondary, "student" model, some pixels are randomly occluded by performing crops in the images. However, the prediction of the *student* network is guided with estimates from the *teacher* network on original images providing a clue even for occluded pixels.

This approach is elaborated even further in **SelFlow** [40]. The main data distillation idea stays the same, but several features are added. Instead of creating occlusions by random crops, which will only lead to out-of-frame occlusions, SelFlow finds superpixels in images and fills them with white noise. It also extends the network from DDFlow to from two to a three-frame setting.

Lastly, **Lai et al.** [38] propose to combine optical flow estimation with stereo matching. The authors create an optical flow estimation network that also serves as a disparity estimator. By using stereo pairs, they are able to add a new loss among the traditional photometric difference between warped target and source images and smoothness losses. They call it a *two-warp loss*, and, as the name suggests, it encourages the similarity of pixels linked through both the optical flow field and the disparity map. They are able to achieve competitive results in depth estimation, but not in optical flow estimation.

2.2 Unsupervised training of multiple tasks

The fact that optical flow on static scene objects is linked to depth and camera motion gave birth to the next two methods. They both combine monocular depth, ego-motion, and optical flow estimation and train all modules in an unsupervised fashion.

GeoNet [36] proposes a two-stage architecture. First, ego-motion and monocular depth estimation are performed. These two estimates are used to create an optical flow field on the rigid parts of the scene. Flow on moving parts is then estimated as a residual motion by the so-called *ResFlowNet*.

The whole architecture is trained in an unsupervised fashion using Structural similarity [5] and pixel brightness to measure the photometric difference between the source frame and target frame warped with the output optical flow. Occlusions are excluded from photometric loss by the same method as in UnFlow [32].

Competitive collaboration [41] extends the whole idea by adding a network for segmentation of static parts of the scene (i.e. motion segmentation). Optical flow on static parts is then estimated from outputs of monocular depth estimation and ego-motion estimation networks together referred to as R. Flow on moving parts is handled by an optical flow network referred to as F.

To train such a complex set of networks in an unsupervised fashion, the authors first pre-train separately R, F and motion segmentation network. Then, they employ a process called *Competitive collaboration.* The process has two phases - in the first, R and F are trained each on the respective parts of a frame as assigned by the motion segmentation network. In the second, motion segmentation is trained based on the comparison of optical flow fields produced by R and F.

2.3 Semi-supervised optical flow

If we do omit cases of unsupervised pre-training and supervised fine-tuning, there were only a few attempts in the optical flow context to create a combination of supervised and unsupervised training.

Xiang et al. [35] and Zhai et al. [44] are two consecutive works that combine the supervised training with some techniques from the unsupervised world. They simply add the photometric consistency and smoothness regularization terms to the supervised loss and attempt to train the network in this way. Unfortunately, they base the network on an outdated FlowNet architecture [19].

Lai et al. [29] present an approach based on a Generative Adversarial Network. The discriminator is trained to recognize the photometric difference map between the source and target image back-warped by either ground truth or estimated optical flow. Further, endpoint error loss is applied alongside the adversarial loss for all labeled data.

The authors show that this semi-supervised setting can lead to slightly better results on all major datasets compared to their supervised baseline. However, the setting assumes that no ground-truth optical flow is available except for the synthetic FlyingChairs [19] dataset.

Chapter 3

Methods

In this chapter, the necessary theoretical foundation is laid for the experiments in work. First, a notation scheme is presented. Afterward, supervised learning is introduced, followed by a more profound introduction to unsupervised learning and its aspects. Lastly, a new semi-supervision method combining the two approaches is proposed.

3.1 Notation

Let I_1, I_2 be two consecutive frames and $\mathbf{f}_{GT,1\to 2}$ ground truth forward optical flow between them.

The goal is to train the parameters Θ of an optical flow estimation network. This is done by minimizing a loss function $L(\Theta)$. The loss function usually has multiple different arguments; however, we omit those to ease the notation.

The architecture used in this work is pyramidal and thus the optical flow estimates are multi-scale i.e. multiple differently-scaled outputs are obtained. Let $l = 1, 2 \ldots 5$ denote the flow pyramid scale from the largest to smallest - $1/4$, $1/8$, $1/16$, $1/32$ and $1/64$ of the input image size.

Let $\mathbf{f}^l_{1\to 2}, \mathbf{f}^l_{2\to 1}$ be the estimated forward and backward optical flow on scale l. By I^l and $\mathbf{f}^l_{GT}$ we denote an image and optical flow down-sampled to scale l respectively[1].

In some sections, flow between three consecutive frames I_1, I_2, I_3 is considered. In this case, we consider I_2 as the reference frame and call optical flow from I_2 to I_1 the *backward* flow $\mathbf{f}^l_{2\to 1}$ and optical flow from I_2 to I_3 the *forward* flow $\mathbf{f}^l_{2\to 3}$.

[1] We use bilinear interpolation for optical flow down-sampling.

3.2 Supervised training

Supervised training works by directly comparing the estimates to some ground-truth data in the loss function. For optical flow, the supervised loss is commonly defined as a standard L2 endpoint-error loss

$$L_{sup}^{L2}(\mathbf{f}_{1\to 2}) = \sum_{l=1}^{5} \alpha_l \sum_{\mathbf{x}\in P} \left\| \mathbf{f}^l_{1\to 2}(\mathbf{x}) - \mathbf{f}^l_{GT,1\to 2}(\mathbf{x}) \right\|_2 . \tag{3.1}$$

where α_l is the pyramid scale weight.

Some experiments feature robust loss function as suggested in [33]

$$L_{sup}^{rob}(\mathbf{f}_{1\to 2}) = \sum_{l=1}^{5} \alpha_l \sum_{\mathbf{x}\in P} \left(\left\| \mathbf{f}^l_{1\to 2}(\mathbf{x}) - \mathbf{f}^l_{GT,1\to 2}(\mathbf{x}) \right\|_1 + \epsilon \right)^q . \tag{3.2}$$

With the default setting $\epsilon = 0.01$ and $q = 0.4$ this metric penalizes the outliers less than the L2 loss.

3.3 Unsupervised training

Unsupervised training avoids the need for direct optical flow ground-truth. Instead, the loss function usually combines multiple terms mainly inspired by the Horn-Schunck method [1, 27]. The main terms are the data term and the regularization term. In short, the data term assures the visual similarity of corresponding pixels. On ambiguities, regularization encourages spatial smoothness. Other terms may be added, such as forward-backward consistency [32].

In this section, the unsupervised loss function is first presented as a whole. Afterward, the data term is explained in more detail, followed by occlusion reasoning methods. Lastly, smoothness regularization term and forward-backward consistency term are defined.

3.3.1 Unsupervised loss function

The unsupervised loss consists of multiple terms that are evaluated on all scales of the network output. In total, in this work, we define four terms. L_D^l is the data term encouraging photometric consistency of the optical flow, L_S^l is the optical flow smoothness term. In the normal two-frame setting, forward-backward consistency term L_C^l is also used.

In the case of three-frame estimation employed solely in experiments with three-frame occlusion reasoning, L_P^l, the combination masks M^l prior is active. Note that in this case, the consistency term L_C^l cannot be used.

The total unsupervised loss is defined as a weighted sum over loss terms and pyramid levels:

$$L_{un} = \sum_{l=1}^{5} \alpha_l (L_D^l + \lambda_S L_S^l + \lambda_C L_C^l + \lambda_P L_P^l), \tag{3.3}$$

where α_l is the pyramid scale weight and λ_S, λ_C and λ_P are weights of the respective loss terms.

3.3.2 Data term

The data term encourages the photometric consistency of estimated optical flow. The definition is based on [32] as follows

$$\begin{aligned} L_D^l(\mathbf{f}_{1\to2}^l, \mathbf{f}_{2\to1}^l) = \sum_{\mathbf{x}\in P} & (1 - o_{1\to2}^l(\mathbf{x}))\rho\Big(f_D(I_1^l(\mathbf{x}), I_2^l(\mathbf{x} + \mathbf{f}_{1\to2}^l(\mathbf{x})))\Big) + \\ & (1 - o_{2\to1}^l(\mathbf{x}))\rho\Big(f_D(I_2^l(\mathbf{x}), I_1^l(\mathbf{x} + \mathbf{f}_{2\to1}^l(\mathbf{x})))\Big) + \\ & \lambda_O(o_{1\to2}^l(\mathbf{x}) + o_{2\to1}^l(\mathbf{x})), \end{aligned} \tag{3.4}$$

where $\rho(x) = (x^2 + \epsilon^2)^\gamma$ (default $\gamma = 0.45$) is the Charbonnier penalty [18] that increases robustness to outliers. Function f_D measures the photometric difference between two pixels.

Variables $o_{1\to2}^l(\mathbf{x})$ and $o_{2\to1}^l(\mathbf{x})$ allow to exclude occluded pixels from the photometric comparison and add a fixed penalty λ_O instead (see Section 3.3.3). If no occlusion handling takes place, consider $o_{1\to2}^l(\mathbf{x}) = o_{2\to1}^l(\mathbf{x}) = 0$.

Photometric difference

The photometric difference function f_D can be defined in many ways. The methods used in this work are presented in the following section.

Brightness difference. The easiest way to measure a photometric difference is to directly compute per-channel intensities difference as

$$f_D^B(I_1(\mathbf{x}_1), I_2(\mathbf{x}_2)) = \sum_{c\in\{r,g,b\}} \|I_2^c(\mathbf{x}_2) - I_1^c(\mathbf{x}_1)\|_1, \tag{3.5}$$

where I^c denotes the color channel c of the image.

However, this naive solution suffers from many pitfalls with the main problems being local ambiguity, non-robustness to illumination changes, or the dependence on the scene illumination (e.g., the differences will be lower for a dark scene compared to a scene using a full brightness range).

Ternary census transform difference. Alternative way to measure the difference is ternary census transform [3, 4]. However, it has to be formulated in a differentiable way for a loss function. We use the definition of [32].

Let I^g be an image converted to grayscale (single channel). Let us define a normalized difference between a pixel $\mathbf{x}$ and its neighbour $\mathbf{x}+\boldsymbol{\delta}$ as

$$D(\mathbf{x},\boldsymbol{\delta}) = \frac{I^g(\mathbf{x}+\boldsymbol{\delta}) - I^g(\mathbf{x})}{\sqrt{\left(I^g(\mathbf{x}+\boldsymbol{\delta}) - I^g(\mathbf{x})\right)^2 + 0.81}}. \tag{3.6}$$

This corresponds to comparing a pixel with its neighbour and assigning a ternary value based on the sign of the difference in the standard Ternary census transform.

Let D_1 and D_2 correspond I_1 and I_2 respectively. The census transform difference is defined as

$$f_D^C(I_1(\mathbf{x}_1), I_2(\mathbf{x}_2)) = \sum_{\boldsymbol{\delta}\in W} \frac{\left(D_1(\mathbf{x}_1,\boldsymbol{\delta}) - D_2(\mathbf{x}_2,\boldsymbol{\delta})\right)^2}{\left(D_1(\mathbf{x}_1,\boldsymbol{\delta}) - D_2(\mathbf{x}_2,\boldsymbol{\delta})\right)^2 + 0.1,} \tag{3.7}$$

where W is a square window of a given size. The above operation corresponds to calculating a Hamming distance between pixels as with standard Census transform.

Census photometric difference is robust to additive and multiplicative changes between two frames since it works with intensities relative to other pixels in a neighborhood instead of with the absolute intensity of one pixel as the brightness difference.

Structural similarity. Another option for f_D is structural similarity measure (SSIM) [5] as demonstrated in [36, 38, 41]. It is specifically designed to measure similarity of two images.

First, let us denote mean and standard deviation of image intensities in a window W around pixel $\mathbf{x}$ computed separately for channel c of image I_i.

$$\mu_i^c(\mathbf{x}) = \frac{1}{|W|} \sum_{\boldsymbol{\delta}\in W} I_i^c(\mathbf{x}+\boldsymbol{\delta}). \tag{3.8}$$

$$\sigma_i^c(\mathbf{x}) = \sqrt{\frac{1}{|W|-1} \sum_{\boldsymbol{\delta}\in W} \left(I_i^c(\mathbf{x}+\boldsymbol{\delta}) - \mu_i^c(\mathbf{x})\right)^2}. \tag{3.9}$$

Mean and standard deviation are used to compare luminance and contrast in the window, respectively. Correlation compares the structural information.

$$\sigma_{12}^c(\mathbf{x}_1,\mathbf{x}_2) = \frac{1}{|W|-1} \sum_{\boldsymbol{\delta}\in W} \left(I_1^c(\mathbf{x}_1+\boldsymbol{\delta}) - \mu_1^c(\mathbf{x}_1)\right)\left(I_2^c(\mathbf{x}_2+\boldsymbol{\delta}) - \mu_2^c(\mathbf{x}_2)\right) \tag{3.10}$$

The original SSIM index is then defined as

$$\mathrm{SSIM}(I_1^c(\mathbf{x}_1), I_2^c(\mathbf{x}_2)) = \frac{\left(2\mu_1^c(\mathbf{x}_1)\mu_2^c(\mathbf{x}_2) + C_1\right)\left(\sigma_{12}^c(\mathbf{x}_1,\mathbf{x}_2) + C_2\right)}{\left(\mu_1^c(\mathbf{x}_1)^2 + \mu_2^c(\mathbf{x}_2)^2 + C_1\right)\left(\sigma_1^c(\mathbf{x}_1)^2 + \sigma_2^c(\mathbf{x}_2)^2 + C_2\right)}, \tag{3.11}$$

where $C_1 = 0.01^2$ and $C_2 = 0.03^2$ are two constants avoiding instability when $\mu_1^c(\mathbf{x}_1)^2 + \mu_2^c(\mathbf{x}_2)^2$ or $\sigma_1^c(\mathbf{x}_1)^2 + \sigma_2^c(\mathbf{x}_2)^2$ is close to zero.

As the aim is to measure the difference between two images instead of similarity, the negative of SSIM is taken. The differences in each channel are summed together:

$$f_D^S(I_1(\mathbf{x}_1), I_2(\mathbf{x}_2)) = \sum_{c \in \{r,g,b\}} \Big(1 - \mathrm{SSIM}\big(I_1^c(\mathbf{x}_1), I_2^c(\mathbf{x}_2)\big)\Big) \tag{3.12}$$

3.3.3 Occlusion reasoning in data term

The data term defined in the previous section suffers from a considerable flaw - pixels from I_1 that are occluded in I_2 cannot be photo-metrically compared and vice-versa. Some methods come up with a way to exclude these pixels from the comparison. In this work, we revise two methods - forward-backward consistency occlusion detection [32] and three frame masking [31].

Occlusion handling by forward-backward consistency

Ideal forward and backward optical flow fields form together a “loop” for pixels that are visible in both I_1 and I_2 i.e., they are forward-backward consistent. However, this generally does not hold for pixels that are occluded in one of the images.

This approximation suffers from many problems - during estimation, there is no ideal optical flow, optical flow is not generally aligned to whole pixels, but it is usually sub-pixel, etc. Nevertheless, [10, 32] propose it for occlusion detection.

In order to allow for small optical flow inaccuracies, a constraint considering also the flow length is defined to detect the occluded pixels:

$$\begin{aligned} o_{1\to2}^l(\mathbf{x}) = \Bigg[\!\!\Bigg[& \left\| \mathbf{f}_{1\to2}^l(\mathbf{x}) + \mathbf{f}_{2\to1}^l(\mathbf{x} + \mathbf{f}_{1\to2}^l(\mathbf{x})) \right\|^2 \\ & \geq \alpha_1 \left(\left\| \mathbf{f}_{1\to2}^l(\mathbf{x}) \right\|^2 + \left\| \mathbf{f}_{2\to1}^l(\mathbf{x} + \mathbf{f}_{1\to2}^l(\mathbf{x})) \right\|^2 \right) + \alpha_2 \Bigg]\!\!\Bigg], \end{aligned} \tag{3.13}$$

where parameters $\alpha_1 = 0.01$ and $\alpha_2 = 0.5$ define the dynamic and static inaccuracy sensitivity and $[\![\cdot]\!]$ is an operator assigning 1/0 if the logical expression inside is true/false. Thus, $o_{1\to2}^l(\mathbf{x}) = 1$ if the pixel on position $\mathbf{x}$ in I_1 is considered occluded in I_2.

Likewise, we define a variable $o_{2\to1}^l(\mathbf{x})$ for backward occlusions with the forward and backward flows swapped in the condition.

Occlusion handling by three frame masking

For general sequences, in three consecutive images I_1, I_2, I_3 almost all pixels from I_2 are visible in either I_1 or I_3. This principle is utilized in [31] to handle occlusions in the following way.

A pixel-wise occlusion mask $M^l(\mathbf{x}) \in \langle 0, 1 \rangle$ is introduced as an additional output of the estimation network. If $M^l(\mathbf{x}) = 1$, the pixel $I_2(\mathbf{x})$ is considered visible somewhere in I_1 and occluded in I_3 and the other way around if $M^l(\mathbf{x}) = 0$. For pixels that are visible in all three images $M^l(\mathbf{x}) = 0.5$ ideally holds.

In this three-frame setting, the backward optical flow $\mathbf{f}_{2\to1}$ is estimated from I_2 to I_1 and forward optical flow $\mathbf{f}_{2\to3}$ from I_2 to I_3. These flow fields are utilized to form an estimate of I_2 denoted as $\tilde{I}_2$ that is created solely from pixels of I_1 or I_3 and guided by M.

$$\tilde{I}_2^l(\mathbf{x}) = M^l(\mathbf{x}) I_1^l(\mathbf{x} + \mathbf{f}_{2\to1}^l(\mathbf{x})) + (1 - M^l(\mathbf{x})) I_3^l(\mathbf{x} + \mathbf{f}_{2\to3}^l(\mathbf{x})). \tag{3.14}$$

Aftewards, the photometric difference is computed between $\tilde{I}_2$ and I_2 as follows

$$L_D^l(\mathbf{f}_{2\to1}^l, \mathbf{f}_{2\to3}^l) = \sum_{\mathbf{x}\in P} \rho\Big(f_D(I_2^l(\mathbf{x}), \tilde{I}_2^l(\mathbf{x}))\Big). \tag{3.15}$$

If the prior assumption holds and thus each pixel from I_2 is visible in at least one of the neighbouring images, this method elegantly avoids evaluating the photometric difference for occluded areas.

Furthermore, a prior is introduced, encouraging the mask to be in the most cases $M^l(\mathbf{x}) = 0.5$ i.e., pixel visible in all three images. This is done by introducing a new term to the loss function

$$L_P^l(M^l) = -\sum_{\mathbf{x}\in P} M^l(\mathbf{x})(1 - M^l(\mathbf{x})). \tag{3.16}$$

3.3.4 Smoothness regularization

Following the same principles as in the Horn-Schunck method [1], smoothness regularization is employed in almost all unsupervised optical flow training methods. We employ the second-order smoothness constraint as in [32], since it has been proved to be beneficial in classical flow estimation methods [6]. To decrease over-smoothing on object edges, we combine it with edge awareness [31].

$$\begin{aligned} L_S^l(\mathbf{f}_{1\to2}^l, \mathbf{f}_{2\to1}^l) = \sum_{\mathbf{x}\in P} \sum_{(\mathbf{s},\mathbf{r})\in N(x)} & \sigma(I_1^l, \mathbf{f}_{1\to2}^l, \mathbf{s}, \mathbf{x}, \mathbf{r}) + \\ & + \sigma(I_2^l, \mathbf{f}_{2\to1}^l, \mathbf{s}, \mathbf{x}, \mathbf{r}), \end{aligned} \tag{3.17}$$

where $N(\mathbf{x})$ contains horizontal, vertical and both diagonal neighborhoods of $\mathbf{x}$ and σ measures the edge-aware optical flow smoothness. The edge awareness is done using first-order spatial difference as the next equation shows.

$$\begin{aligned}\sigma(I,\mathbf{f},\mathbf{s},\mathbf{x},\mathbf{r}) = &\exp\big(-\|I(\mathbf{x})-I(\mathbf{s})\|_2\big)\exp\big(-\|I(\mathbf{x})-I(\mathbf{r})\|_2\big)\cdot \\ &\cdot\boldsymbol{\rho}(\mathbf{f}(\mathbf{s})-2\mathbf{f}(\mathbf{x})+\mathbf{f}(\mathbf{r})).\end{aligned} \tag{3.18}$$

We assume $\boldsymbol{\rho}(\cdot)$ computes the average over the penalties from x- and y-components

$$\boldsymbol{\rho}(\mathbf{p}) = \frac{1}{2}(\rho(p_x)+\rho(p_y)). \tag{3.19}$$

3.3.5 Forward-backward consistency term

As it is already described in Forward-backward consistency occlusion handling part (Section 3.3.3), forward and backward optical flow fields between two images $\mathbf{f}^l_{1\to2}(\mathbf{x})$ and $\mathbf{f}^l_{2\to1}(\mathbf{x})$ form "loops" if the pixel is visible in both I_1 and I_2. As it is proposed in [32], we can directly encourage this consistency by adding the following term to the loss function.

$$\begin{aligned}L^l_C(\mathbf{f}^l_{1\to2},\mathbf{f}^l_{2\to1}) = \sum_{\mathbf{x}\in P}&\boldsymbol{\rho}\Big(\mathbf{f}^l_{1\to2}(\mathbf{x})-\mathbf{f}^l_{2\to1}(\mathbf{x}+\mathbf{f}^l_{1\to2}(\mathbf{x}))\Big)(1-o^l_{1\to2}(\mathbf{x}))+ \\ &+\boldsymbol{\rho}\Big(\mathbf{f}^l_{2\to1}(\mathbf{x})-\mathbf{f}^l_{1\to2}(\mathbf{x}+\mathbf{f}^l_{2\to1}(\mathbf{x}))\Big)(1-o^l_{2\to1}(\mathbf{x})),\end{aligned} \tag{3.20}$$

where $o^l_{1\to2}(\mathbf{x})$ and $o^l_{2\to1}(\mathbf{x})$ allow to exclude occluded pixels if occlusion masking is active. Otherwise, consider $o^l_{1\to2}(\mathbf{x}) = o^l_{2\to1}(\mathbf{x}) = 0$.

3.4 Semi-supervised training

Semi-supervised training attempts to combine supervised and unsupervised approaches and obtain the best from both worlds. This section first describes a naive approach to semi-supervision, then we propose a method that constrains the updates from unsupervised loss by a supervised gradient.

3.4.1 Naive semi-supervision

The naive semi-supervision is formulated as e.g. in [35] simply as a combination of supervised and unsupervised terms

$$L_{comb} = L^{L2}_{sup} + \lambda_U L_{un}, \tag{3.21}$$

where L^{L2}_{sup} is the L2 endpoint-error supervised loss (Eq. 3.1), L_{un} is the unsupervised loss (Eq. 3.3) and λ_U is the unsupervised loss weight.

3.4.2 Constrained semi-supervision

The previous naive approach may lead to disturbances in training - either the effect is negligible for small λ_U, or we risk losing the performance of supervised training with high λ_U. In order to minimize the disturbances caused by the introduction of unsupervised loss to the training, we propose the following approach.

At each iteration during training, the network is evaluated on one pair of frames with the ground-truth optical flow (supervised sample) and N pairs without ground-truth (unsupervised samples). Respective supervised and unsupervised loss functions are evaluated for each sample separately. Then, by back-propagation, we compute the per-sample network parameter update gradient.

Afterward, the gradient from the supervised sample is used to pose a constraint on the unsupervised gradients. All unsupervised updates that might increase L_{sup} and thus potentially lead to worse performance are skipped.

For the supervised sample, the gradient is defined as

$$\mathbf{G}_s = \nabla L_{sup}(\Theta) \tag{3.22}$$

and for i-th unsupervised sample the gradient is

$$\mathbf{G}_u^i = \nabla L_{un}(\Theta). \tag{3.23}$$

$\mathbf{G}_s$ is used as the constraining vector. Positive dot product with the constraining vector ensures that the added $\mathbf{G}_u^i$ does not have an orientation opposite to $\mathbf{G}_s$. Thus, the parameter update vector is defined as:

$$\mathbf{G} = \mathbf{G}_s + \sum_{\forall i: \mathbf{G}_u^i \cdot \mathbf{G}_s > 0} \lambda_M \mathbf{G}_u^i, \tag{3.24}$$

where λ_M is the unsupervised gradient weight.

Because G_s is the gradient of L_{sup} at Θ and because we added only such $\mathbf{G}_u^i$ that fall in a half-space defined by G_s, by updating the parameters by $-\mathbf{G}$, the value of L_{sup} linearized at Θ would not rise.

Chapter 4

Experiments

This chapter describes experiments conducted with unsupervised and semi-supervised training and introduces technical details. First, the estimation network architecture is introduced along with the used datasets. Next, we focus on unsupervised training - starting with a simple photometric difference measures comparison and then transferring to a more evolved experiments with occlusion reasoning or dataset size. The gained insight is then utilized in experiments with semi-supervised training. Lastly, technical details like evaluation metrics and training parameters are included.

4.1 Preliminaries

The popular PWC-Net architecture is selected for the presented experiments for its competitive performance combined with simplicity. It is described in this section in more detail. Next, we provide a quick overview of the datasets used throughout this work.

4.1.1 Network architecture: PWC-Net

PWC-Net [33] is a popular optical flow estimation network combining three main ideas - image features Pyramid, feature Warping and Cost volume - hence the name PWC. Figure 4.1 shows a diagram of the architecture.

A pyramidal approach is used for optical flow estimation. The individual levels of the pyramid $l = 1, 2 \ldots 5$ work with $1/4$ to $1/64$ of the original image resolution. The estimation is done gradually, starting with the smallest scale. Thanks to this approach, coarse movements are detected on smaller scales and then only refined higher up the pyramid. The optical flow output on the highest level $\mathbf{f}^1_{1\to 2}$ is $1/4$ of the input image size and hence has to be interpolated to the original resolution.

Name	Train pairs	Test pairs
Creative Flow+ [42]	153298	9506
Flying Chairs [19]	22232	640
KITTI 2012 [12]	6572*	194
KITTI 2015 [25]	6800*	200
KITTI raw [25]	95562*	-
Sintel Clean [11]	781	87
Sintel Final [11]	781	87
Sintel movie [8]	9372*	-

Table 4.1 Overview of optical flow datasets used in the experiments. A star (*) denotes that no ground-truth optical flow for training is available.

As standard in recent neural networks, images on the input are first fed to an extractor (aka encoder). This is a series of convolutional layers that output so-called image features E^l on the corresponding scales. These features have many channels, ranging from 32 for $l = 1$ to 196 for $l = 5$. Note that the same convolution weights are used on I_1 and I_2.

On the smallest scale, $l = 5$, image features are correlated, and a cost volume CV^5 is produced. This means that for each pixel in E_1^5, a scalar multiplication (in the channel dimension) with pixels from a corresponding 9×9 neighborhood in E_2^5 is done. The resulting cost volume is fed into an optical flow estimator. This block has several layers of convolutions producing two outputs - so-called optical flow features F^5 and optical flow $\mathbf{f}_{1\to2}^5$ itself. Afterwards, F^5 and $\mathbf{f}_{1\to2}^5$ are upscaled using deconvolution (aka transposed convolution) to the next higher scale. Note that the optical flow estimators on different scales do not share weights.

On the next scale, $l = 4$, the upscaled optical flow estimate is used to warp the extractor features E_2^4 from the second image. Thus, the following correlation CV^4 only refines the coarse flow estimates from the previous level. The same schematic is repeated, as shown in Figure 4.1. On the largest scale, there is also a refining module that enhances the optical flow right before the output.

4.1.2 Datasets

This section lists all datasets that are used for training and for testing in the experiments. Table 4.1 lists a quick overview of the number of train and test image pairs for each dataset.

Sintel [11] is a current standard benchmark for optical flow evaluation. The dataset includes forward optical flow ground-truth along with occlusion ground-truth. It is derived from the open source 3D animated short movie *Sintel* [8]. Two render passes are used for testing - basic *Clean* and *Final* that adds motion blur, atmospheric effects and other.

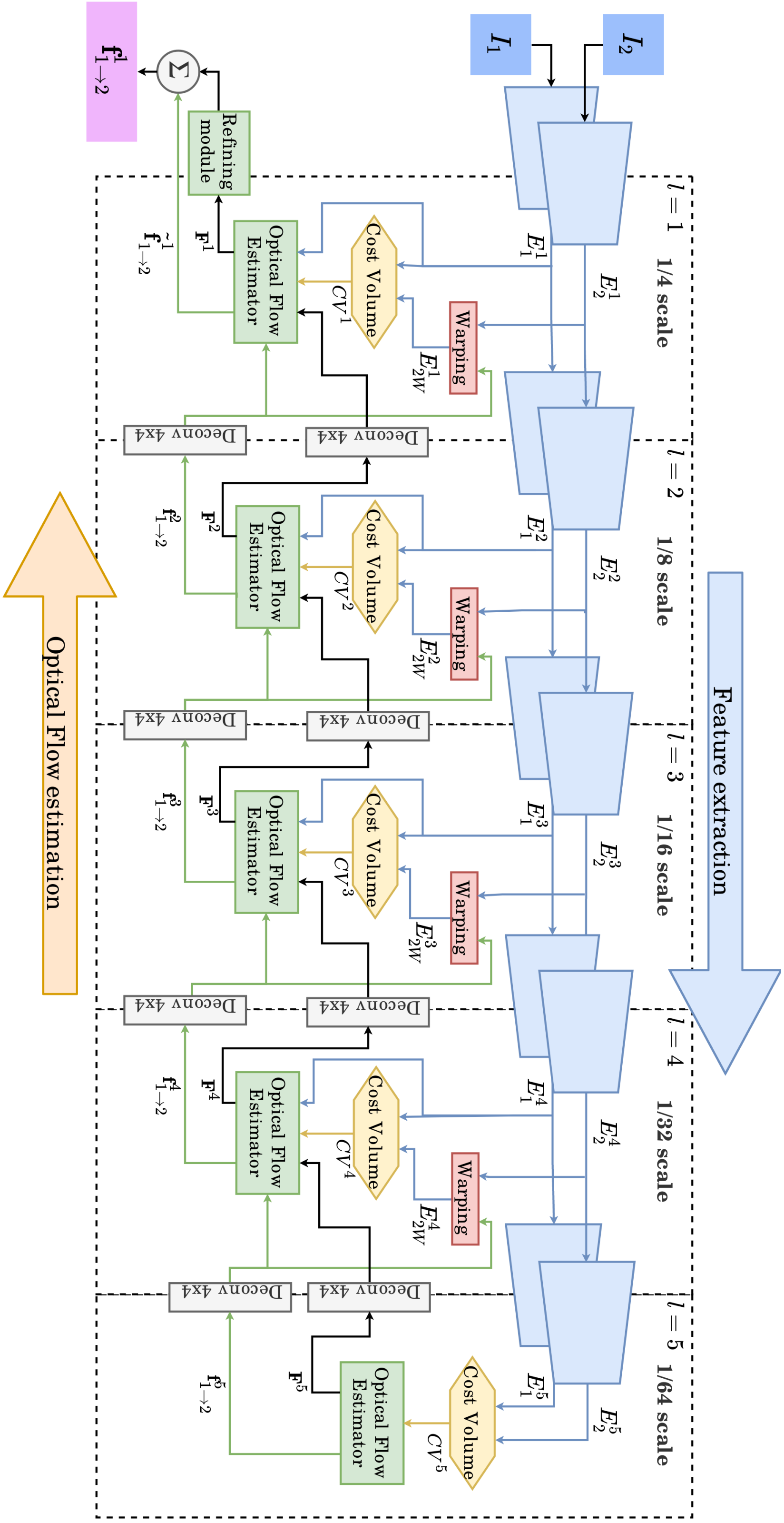


Figure 4.1 PWC-Net architecture.

To avoid complicated evaluation, a 90-10 split of the publicly-available data to training and testing parts is created yielding 1562 train and 2 × 87 test samples (separately clean and final pass). In training, both *Clean* and *Final* passes are combined.

Sintel movie. All frames from the original movie [8] were extracted for unsupervised and semi-supervised training, similarly to [40]. To cope with compression artifacts, we downscaled the 4K resolution images to 1152 × 648. Cuts between scenes, where no optical flow exists, were avoided with PySceneDetect [37]. Moreover, too dim (typical for fade-ins/outs) or too similar consecutive images were detected using pixel-wise brightness resp. brightness difference and excluded. Altogether, 9372 samples were created.

KITTI is a dataset featuring sequences a from car front camera with available sparse optical flow ground truth. It comes in two editions: 2012 [12] and 2015 [22]. The latter includes optical flow on some objects that are moving in the scene, which increases the complexity. The whole publicly available annotated parts of the datasets are used for testing. Frames from the multiview extensions (i.e., frames before/after the annotated pair) are used for unsupervised/semi-supervised training while excluding both frames from the annotated pair.

Moreover, KITTI raw [14] collects a large amount of unlabeled data that was recorded during the creation of the KITTI dataset. In some semi-supervised experiments, this ca. 95K sample dataset is used.

Creative Flow+ (CF+) [42] is a recently introduced dataset with artistic-like scenes and ground truth optical flow. The computer rendered scenes mimic styles found in a variety of animated movies - uniform colored surfaces, objects with changing texture, different outlines, etc. Tests are done on the 10K sample list provided by the authors. Some of the experiments also utilize the set of 153K *mixamo* train frames. Full resolution images (1500 × 1500) are used. Note that with CF+, it is more meaningful to observe performance on the foreground areas since optical flow on the background is often not well defined.

FlyingChairs [19] is a synthetic dataset designed for pre-training optical flow networks. It features 3D models of chairs performing random movements over a picture background. No illumination changes nor motion blur is present, making it ideal for pre-training of our unsupervised models.

4.2 Unsupervised learning

In this section, we revise the main recently proposed ideas in unsupervised learning and conduct experiments under a unified setting making the results comparable. We first focus on photometric difference measures and occlusion reasoning. Forward-backward consistency term experiments then follow.

Lastly, we examine the connection between the training dataset size and optical flow estimation accuracy.

4.2.1 Photometric difference measures

In order to compare photometric difference measures, three models are trained with unsupervised loss L_{un} (Eq. 3.3). Starting with a common FlyingChairs pre-trained model (see Technical details in sec. 4.5.1), the photometric difference function f_D is set in the further training as either per-channel brightness difference, ternary census transform difference or structural similarity (see Section 3.3.2).

Following [32] the regularization weight with brightness difference is set to $\lambda_S = 3.0$ (higher setting was also tested with a negative result). Out of three tested candidates, the same $\lambda_S = 3.0$ is also found to perform the best with census photometric difference. With SSIM, we observe an average magnitude of the data term and set λ_S so that the ratio between data and regularization terms is similar to the previous experiments i.e., $\lambda_S = 0.1$.

No other terms besides data and smoothness are active during this series of experiments. Models are trained on Sintel and KITTI training data until convergence.

4.2.2 Occlusion reasoning

To test the two presented occlusion reasoning methods - occlusion handling by forward-backward consistency and three-frame masking (Section 3.3.3) - a series of experiments is conducted. First, several scenarios are tested for both methods starting from the common FlyingChairs pre-trained model (Section 4.5.1). Afterward, introducing occlusion handling to an already-trained model from the previous experiment is tested.

Occlusion handling by forward-backward consistency

First, the effect of occlusion reasoning by forward-backward consistency check on the estimated optical flow is examined. As in the previous experiment, we start with an (unsupervised) FlyingChairs pre-trained model and continue training with loss function L_{un} (Eq. 3.3) on KITTI and Sintel datasets. Again, the regularization term is set $\lambda_S = 3$. However, the pixels detected as occluded (as described in Section 3.3.3) are masked out. Three λ_O penalty settings are tested $\lambda_O = 8$, $\lambda_O = 35$ and $\lambda_O = 70$.

The per-channel brightness difference is set as f_D. Census photometric difference is only tested with $\lambda_O = 8$ and $\lambda_O = 70$, because the experiments show no significant differences between the individual settings of λ_O.

Occlusion handling by three-frame masking

Next, the effect of occlusion reasoning by combining images I_1 and I_3 to form an estimate of I_2 (see Section 3.3.3) is explored. Note that in this setting we compute the backward optical flow $\mathbf{f}^l_{2\to1}$ from I_2 to I_1 and the forward flow $\mathbf{f}^l_{2\to3}$ from I_2 to I_3.

MaskNet integration. In order to produce the combination mask M^l, the internal results in PWC-Net are collected at each scale and fed into our convolutional neural network called MaskNet, as shown in Figure 4.2. PWC-Net runs between frames I_2, I_1 and I_2, I_3 simultaneously. At each scale, the cost volume from each run CV^l_1 and CV^l_3 along with warped image features E^l_{1W} and E^l_{3W} are fed into MaskNet producing M^l. Comparing the cost volumes leads to a decision if a pixel is occluded in either image I_1 or I_3 - the correlation magnitude will presumably be lower if an occlusion takes place. Image features E^l_{1W} and E^l_{3W} serve as an additional cue.

MaskNet is a simple network consisting of five 3×3 convolutional layers, each followed by the LeakyReLU activation function except for the last one, which is followed by Softmax function. Thus, the output has two channels - one is M^l, the other $(1 - M^l)$. MaskNet convolution weights are shared on all scales.

Training and testing. The training datasets are changed to work with three consecutive images instead of two, and thus, the size is slightly decreased. Since the tested occlusion reasoning method helps only during training, optical flow estimation testing is done on the common test frame pairs. Occlusion estimation is evaluated only on the Sintel dataset, where dense occlusion ground-truth is available. For this particular evaluation, the test frame pairs had to be extended with one frame from the training set to form test triplets.[1]

Since the occlusion annotation is only available in forward flow i.e., from I_2 to I_3, the occlusion estimation is tested only in this direction. We assume that $M^l(\mathbf{x}) = 0.5$ designates the ideal situation when the pixel $\mathbf{x}$ is visible in both I_1 and I_3. Thus, only values greater than 0.5 are interpreted as an occlusion in I_3. Two methods are used for the testing. In the first, the M^l values are mapped from interval $\langle 0.5, 1\rangle$ to interval $\langle 0, 1\rangle$ and interpreted as a probability of occlusion. Precision-recall curve and measure AUC (area under curve) is then calculated. The other way of testing establishes a threshold - specifically, we test 0.75 - and considers all pixels $M^l(\mathbf{x}) > 0.75$ as occluded. Precision and recall values are then measured.

Experiments. As in the previous experiments, training starts with the pre-trained FlyingChairs model and works with the unsupervised loss L_{un} (Eq. 3.3). MaskNet weights are initialized randomly. The data term is calculated according to Eq. 3.15, i.e. respecting the three-frame occlusion

[1] Since the training is unsupervised; we presume that the effect of such mixing is negligible.

reasoning. Since the photometric error is measured only between one pair of images instead of two, the magnitude of the data term is halved and thus the regularization term[2] weight has to be halved to $\lambda_S = 1.5$. Training is done on the combination of KITTI and Sintel datasets.

Initial experiments are done with brightness difference as the f_D photometric measure with two λ_P mask prior settings $\lambda_P = 0$ and $\lambda_P = 5$. As these fail to decrease the test error w.r.t baseline with no occlusion reasoning, further experiments are done with census photometric difference measure with a broader range of λ_P settings.

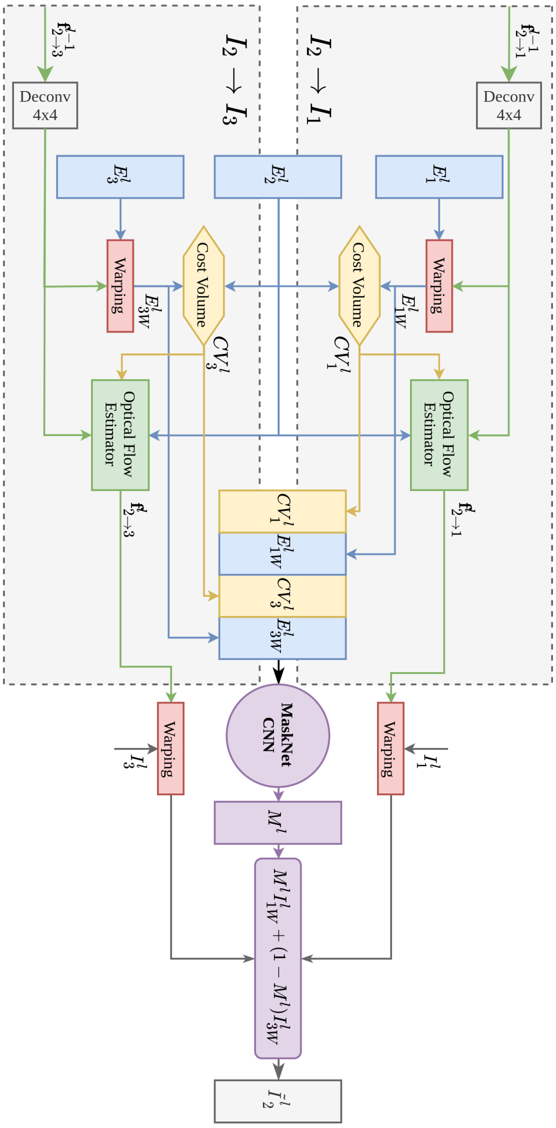


Figure 4.2 MaskNet integration to the PWC-Net architecture allowing for three-frame occlusion reasoning. One level of the estimation pyramid is displayed. M^l allows to select pixels from I_1 or I_3 that are occluded in occluded in the other image.

[2] Note that still, two optical flow fields are regularized: $\mathbf{f}^l_{2\to1}$ and $\mathbf{f}^l_{2\to3}$.

Fine-tuning with occlusion reasoning

As the previous experiments are unable to bring significant improvement, a scenario with network fine-tuning is put to the test. We start with a pre-trained model from the photometric difference measures experiment (Section 4.2.1) - specifically, the one trained with census photometric difference as it achieved the best accuracy. Training is continued with the same photometric measure for another 100 epochs (i.e., ca. 350K iterations) with learning rate starting at 1e-5 and halving every 50K iterations. The datasets remain unchanged - Sintel and KITTI.

However, occlusion reasoning is added. The idea behind is that since the model already performs well in optical flow estimation, occlusions will also be estimated better, and thus the occlusion reasoning in data term will have a more significant effect.

We establish one control experiment with no occlusion reasoning. Two are set with forward-backward occlusion masking with penalties $\lambda_O = 8$ and $\lambda_O = 70$. This setting is chosen based on the results of the previous experiments. Another two experiments are done with three-frame occlusion reasoning - one without the mask prior $\lambda_P = 0$, the other with $\lambda_P = 50$. The regularization weight λ_S is set as in the respective previous sections.

4.2.3 Forward-backward consistency term

To test the effect of forward-backward consistency term (Section 3.3.5), two models are trained with unsupervised loss (Eq. 3.3) and $\lambda_C = 0.3$ (this setting is presented as the best in [32]) - one with brightness difference and the other census difference as f_D. Another two models with settings $\lambda_C = 0.5$ and $\lambda_C = 3.0$ are trained only with census difference.

The other settings are shared with the previous experiments - weights are initialized from FlyingChairs pre-trained model, training is done on Sintel, and KITTI datasets and smoothness regularization is kept at $\lambda_S = 3$.

4.2.4 Dataset size experiments

One of the arguments for unsupervised training is that it can work with virtually unlimited data. Since there is no need for annotation, any video can be added to the training set.

On the other hand, the unsupervised loss formulates an objective of photo-consistent and smooth optical flow, which is more universal than a supervised objective where only output optical flow examples are given. This suggests that even small dataset might be enough for training the unsupervised objective, because the underpinning principle is explicitly formulated and does not have to be searched for in examples.

In this section, we experiment with the training dataset size. For the experiments, we use the same setting as in photometric difference measures experiments (Section 4.2.1) with brightness difference as f_D so that it can serve as the baseline.

Big dataset

To test whether a large dataset improves results with unsupervised training, we collect a big number of training samples from publicly available sources. Table 4.2 lists an overview. The dataset is intended to have two equally large parts (each ca. 100K pairs) corresponding to the type of test data: KITTI-like part and a Sintel-like part.

Sequences on a YouTube channel "J Utah" feature many different high-resolution car front-cam videos. Various locations and scenarios (light and weather conditions) are captured. We select several of these sequences, as listed in Table 4.2 and references. They serve as the part similar to the KITTI dataset.

To construct the Sintel-like part, we include open-source videos from projects similar to Sintel: Caminandes (all three episodes) [15, 16, 26], Cosmos laundromat [21], Tears of steel [13]. We also include the Sintel movie dataset that was already described (Section 4.1.2). Besides, we also add three compilations of movie trailers acquired from YouTube [50–52]. The genre was chosen because, unlike whole movies, trailers usually tend to contain scenes with much more movement.

The presented data are collected at the highest available quality and processed in the same way as Sintel movie dataset (Section 4.1.2) - frames are downscaled to 1152×648, cuts between scenes are detected by PySceneDetect [37] and too dim or too similar consecutive images are excluded.

To mimic the baseline, training is done with the unsupervised loss (Eq. 3.3), weights are initialized from FlyingChairs pre-trained model, and smoothness regularization is set to $\lambda_S = 3$. The above-presented dataset is combined with standard KITTI and Sintel data to form the training set. The learning rate starts at 1e-4. Since the standard 100K iterations interval is not enough to observe a plateau on the training loss, we halve the learning rate after 250K, 600K, 950K, 1521K, 1571K, and 1621K iterations. Altogether, 1764K iterations (ca. 32 epochs) are performed. Note that the model takes significantly longer to converge.

Small dataset

To test how the estimation network behaves when trained with an unsupervised loss on a small amount of data, twenty samples are collected from each

Name	Type	# samples
J Utah seqs. [45–49]	driving	101783
Caminandes (1-3) [15, 16, 26]	animated	5618
Sintel movie [8]	animated	9893
Cosmos laundromat [21]	animated	12408
Tears of steel [13]	anim.+live	13672
Trailers [50–52]	anim.+live	62909
Total		*206283*

Table 4.2 Summary of collected data for the big dataset experiment.

subset of common training datasets - Sintel clean, Sintel final, KITTI 2015 and KITTI 2012 i.e., 80 frame pairs in total.

For training, the model weights are initialized randomly (i.e., no pre-training phase). Training takes 912K iterations - roughly the same number of iterations as pre-training plus main training of the baseline experiment. The learning rate is initiated at 1e-4 and halved every 100K iterations. As in the baseline, brightness photometric difference is used as f_D and $\lambda_S = 3$.

4.3 Semi-supervised training on single/close domain

In this section, we experiment with the proposed semi-supervision method (See 3.4.2) on Sintel and KITTI datasets. First, semi-supervised fine-tuning is tested. We attempt to increase the accuracy of a model supervised on Sintel by adding unsupervised samples from the Sintel movie dataset. Second, domain adaptation of the supervised model to KITTI is tested.

To start with, a model is trained on Sintel with supervised loss function. We refer to this model as “Sintel supervised model”. It is trained with L2 endpoint-error loss L_{sup}^{L2} (Eq. 3.1) on the Sintel dataset. The weights are initialized from the common FlyingChairs pre-trained model (see Section 4.5.1). The learning rate starts on 1e-4 and is halved when we observe a plateau in training loss - after 100K, 170K, 240K iterations. Convergence is reached after 260K iterations.

4.3.1 Semi-supervised fine-tuning

The goal of semi-supervised fine-tuning is to improve the performance of the “Sintel supervised model” on the Sintel dataset by training on unlabeled data. We thus start with the “Sintel supervised model” and train with the proposed constrained gradient semi-supervision method. Sintel dataset serves as the supervised domain, and frames from the Sintel movie dataset serve as the unlabeled samples.

The L2 endpoint error L^{L2}_{sup} stays at the place of the supervised loss. The unsupervised loss utilizes census transform difference in the data term f_D and sets $\lambda_S = 3$ as in the previous unsupervised experiments. In one iteration, six unsupervised samples are fed to the semi-supervision method.

During experiments, we discovered that the switch to semi-supervised training leads to a small jump in the test error. Two schemes are tested to cope with the effect. In the first, "short", the jump is minimized by setting low learning rate and unsupervised gradient weight λ_M. The second, "long" allows for a higher jump that is compensated with more extended training.

The "short" scheme sets $\lambda_M = 0.1$. The learning rate starts with 1e-5 and is halved after 20K, 35K, 45K, 50K, 52.5K, and 53.75K iterations. Training takes 54K iterations. "Long" sets $\lambda_M = 0.3$. The learning rate is initialized to 2e-5 and halved after 50K, 75K, 87.5K, and 93.75K iterations. It takes 101K iterations.

4.3.2 Semi-supervised domain adaptation

Domain adaptation aims to discover whether the performance on a close domain can be improved by adding unsupervised samples from a close domain. We thus combine supervised samples from Sintel and unsupervised samples from the KITTI raw dataset with the proposed semi-supervision method. The experiments are designed similarly to previous fine-tuning section - the training starts with the "Sintel supervised model", combined supervised and unsupervised loss functions are the same and both "short" and "long" schemes are tested.

4.4 Semi-supervised training on distant domain

In the next experiments, the possibility of semi-supervised distant domain adaptation is explored. A model is trained with supervision for a certain domain, but due to generalization issues, it fails to produce accurate results on some distant domain. The goal is to improve its performance on the distant domain while maintaining the performance on the original domain. Ideally, this is done with no ground-truth optical flow from the distant domain as it is generally not available.

The recently published Creative Flow+ dataset (CF+) [42] serves as a good example of a possible distant domain as it features various artistic-like scenes. The authors show that all Sintel-trained CNN-based estimators fail to generalize on this domain.

This section is structured as follows. First, a supervised model is fine-tuned using the semi-supervision method to combine Sintel and Sintel movie datasets and the accuracy on CF+ is observed. Second, the same setting is repeated, but now featuring frames from CF+ as the unsupervised samples. Then, to check whether constraining is necessary, the semi-supervision is posed as a naive unconstrained loss combination (see Section 3.4.1). Lastly, we establish a baseline supervised on both Sintel and CF+ to have a fully-supervised comparison.

Note that unlike the previous series of experiments (Section 4.3), the semi-supervision experiments on a distant domain work with pre-trained PWC-Net model made available by the authors [33]. The model was trained on FlyingChairs and FlyingThings3D [25] datasets; and then fine-tuned for the Sintel dataset. It achieves better performance on the Sintel dataset than our supervised models thanks to a more evolved training process. Let us refer to the Sintel fine-tuned model as "PWC-Sintel". To keep the training protocol, all training staring with the "PWC-Sintel" model works with 768 × 384 image resolution.

4.4.1 Semi-supervision on single domain

In the first experiment, similarly to Section 4.3, a model is trained with the constrained semi-supervised method combining the supervised samples from Sintel and unsupervised from Sintel movie.

The training starts with "PWC-Sintel" model. Experiments are done with both census and brightness photometric differences as f_D in the unsupervised loss. For other unsupervised terms, we pick the best consistently performing setting $\lambda_S = 3.0, \lambda_C = 0.3$. Both L2 endpoint-error L_{sup}^{L2} and robust error L_{sup}^{rob} are tested as the supervised loss function. As in the previous semi-supervised experiments, six unsupervised samples are fed at each iteration.

Again, when the training starts a test error jump is observe. To minimize the effect, we propose the following technique. The optimization is warmed-up by performing three epochs solely with a supervised loss. They are followed by 1-2 semi-supervised epochs (depending on the learning rate schedule) with a small learning rate 1e-7.

Afterward, two training schemes that are slightly different from previous experiments are employed. "Short" tries to utilize the observation from the unsupervised experiments that halving the learning rate leads to a sharp test error decrease. The schedule is 56K iterations long and halves the learning rate after 20K, 35K, 45K, 50K, 52.5K, 53.75K, 55K iterations. "Long" schedule takes 135K iterations and the learning rate is halved after 30K, 50K, 70K, 90K, 105K, 120K. Both schedules set $\lambda_M = 0.1$. We find this setting to be enough to promote the unsupervised loss and, at the same time, minimize the test error jump. The initial learning rate is also a point of experimentation - we test 1e-5, 1e-6, and 1e-7.

In order to establish a control experiment, we also continue with supervised training on Sintel using L2 endpoint-error L_{sup}^{L2}. Model weights are initialized from "PWC-Sintel", the learning rate starts at 1e-5 and is halved according to the "Long" schedule.

4.4.2 Semi-supervision including distant domain

Next, the idea of the previous experiment is further developed by including unlabeled samples from the distant domain. The network is trained in the same way as in the previous experiment, with the only difference that the unsupervised samples are taken from the training part of the CF+ dataset (i.e., frames only, no GT flow).

Based on the results from previous experiments, only brightness difference is tested as f_D as it seems to perform better with CF+. Both "Short" and "Long" schedules are examined.

4.4.3 Unconstrained semi-supervision

To test the need for the constrained semi-supervision method, experiments without any constraining take place. The loss is naively defined as a combination of supervised and unsupervised terms, as introduced in Section 3.4.1.

As in previous sections, the experiments start with the "PWC-Sintel" model. The network is trained with L_{comb} as a loss function (Eq. 3.21) on the Sintel dataset. We test three settings of λ_U; $\lambda_U = 0.1, \lambda_U = 1$ and $\lambda_U = 2$. The parameters of the unsupervised loss remain the same as in constrained semi-supervision experiments, i.e. $\lambda_S = 3.0, \lambda_C = 0.3$, and a brightness difference measure is selected as f_D.

During all three experiments, a CF+ test error drop occurs in the ca. first 30K iterations. However, the test error then rises even above the control experiment (supervised training on Sintel, see end of Section 4.4.1). At the same time, both terms of the loss L_{sup} and L_{un} steadily decrease during the whole training. This suggests that L_{comb} leads to an over-fitting of the unsupervised objective on Sintel and prevents to generalize it on CF+. For each experiment, we state the situation before the CF+ test error rise in the final results.

4.4.4 Supervised training

To establish a fully-supervised comparison, we also fine-tune the "PWC-Sintel" model for the CF+ dataset using the ground truth optical flow. In each training epoch, we train on all Sintel training samples and the same number (i.e., 1562) of randomly chosen CF+ samples. L2 endpoint-error

L_{sup}^{L2} is chosen as the supervised loss. Training takes 171K iterations starting with learning rate 1e-5 that is gradually halved with a “Long” schedule.

4.5 Common technical details

This section lists all the technical details regarding training. All experiments share the same settings unless stated otherwise. Afterward, the employed optical flow evaluation metrics are described.

4.5.1 Training settings

For training, Adam optimizer [17] is used with $\beta_1 = 0.9, \beta_2 = 0.999$. Batch size is four. As it is common in other works [32, 39], learning rate is initialized at 1e-4 and halved every 100K iterations. The training is always run until a convergence is observed.

As in original PWC-Net paper [33], the pyramid weights are $\alpha_1 = 0.005$, $\alpha_2 = 0.01$, $\alpha_3 = 0.02$, $\alpha_4 = 0.08$, $\alpha_5 = 0.32$. Network weights are initialized using He (aka Kaiming) initialization [20].

The resolution of training frames is 896×320 - the original frames are randomly cropped to this size. As of other augmentations, both common and relative (between frames in a pair/triplet) geometric transforms are used: random rotation, translation, scale, squeeze and flip. Photometric transforms are also included: random gamma, brightness, contrast, and relative color channel brightness changes.

Census photometric difference is computed on different window sizes at each pyramid scale, from the largest to the smallest scale it is: $7 \times 7, 7 \times 7, 5 \times 5, 3 \times 3, 3 \times 3$. We found that with Census transform, this progressive sizing leads to slightly better results than a fixed size of 3×3. The structural similarity is found to work the best with the constant window size of 3×3.

Pre-trained unsupervised model.

To cope with convergence problems of models[3] a common pre-trained model is established. This model serves as initialization for experiments. Training is done on the FlyingChairs dataset (see Section 4.1.2), with unsupervised loss function L_{un} (Eq. 3.3). The setting is kept as simple as possible, and thus brightness difference is chosen as f_D, $\lambda_S = 3$, $\lambda_C = 0$, and no occlusion reasoning is active. The convergence is observed after 240K iterations.

[3] In our first experiments, randomly initialized models did not converge when training with census photometric difference or with occlusion reasoning.

4.5.2 Optical flow evaluation

EPE refers to an average endpoint error, a standard optical flow error measure. We calculate it as

$$\frac{1}{\sum_{P \in S} |A(P)|} \sum_{P \in S} \sum_{x \in A(P)} \left\| \mathbf{f}^P_{1 \to 2}(\mathbf{x}) - \mathbf{f}^P_{GT,1 \to 2}(\mathbf{x}) \right\|_2 , \tag{4.1}$$

where S is a set of test samples P, $A(P)$ defines the area of interest (whole image, foreground pixels etc.) and $\mathbf{f}^P_{1 \to 2}$ is the flow estimated on sample P scaled to original image size (we use bilinear interpolation).

Fl-all is an error measure proposed for the KITTI dataset, where there is an uncertainty in optical flow measurements. It is defined as the percentage of optical flow outliers, i.e., flow end-point error is $> 3px$ and $> 5\%$.

Optical flow visualization is done with color-wheel. The vector angle is coded by hue and its length by saturation of the color. Figure 4.3 shows the visualisation key.

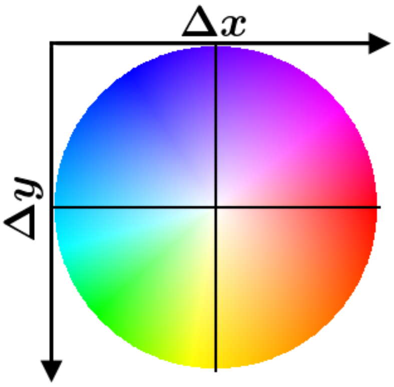


Figure 4.3 Optical flow visualization key.

Chapter 5

Results and discussion

The next chapter presents results from the experiments and provides a discussion. The structure from the previous section is preserved - first, we focus on unsupervised training and its various aspects. Afterward, semi-supervision experiments are analyzed.

5.1 Unsupervised training

This section revises the results from experiments concerning unsupervised training. First, photometric difference measures and occlusion reasoning methods are examined. Next, other aspects are discussed - forward-backward consistency term and the dataset size.

5.1.1 Photometric difference measures

The results from the experiments with different photometric measures are listed in Table 5.1. Census transform seems to behave the best on both Sintel and KITTI domains. SSIM seems to lie somewhere between brightness and census measures.

The performance gap between census and brightness difference is significantly larger on KITTI. The situation depicted in Figure 5.1 can provide a potential explanation. Usually, big parts of images in the KITTI dataset are filled with road texture, which is very ambiguous. As the figure shows, brightness difference seems to fail on this ambiguity - optical flow outliers cover a large part of the road. However, since SSIM and Census transform compare a local structure around the given pixels, the optical flow estimates on the road surface are much more precise.

In the figure, we can also observe the robustness of Census difference measure to illumination - whereas SSIM fails in the dim right-bottom corner of the image, Census is not affected.

	Sintel AEPE [px]						KITTI	
	Clean			Final			Fl-all [%]	
Metric	**ALL**	**NOC**	**OCC**	**ALL**	**NOC**	**OCC**	**2015**	**2012**
Brightness	5.34	3.38	31.65	6.41	4.35	33.99	41.94	29.74
Structural similarity	4.67	2.82	29.44	5.73	3.76	32.13	29.72	17.39
Census transform	4.08	2.31	27.74	5.24	3.36	30.54	26.35	13.25

Table 5.1 Photometric difference measures comparison. Average endpoint error on Sintel and KITTI datasets for the tested photometric difference measures. ALL/NOC/OCC refer to all/non-occluded/occluded pixels. Fl-all is outlier ratio for KITTI.

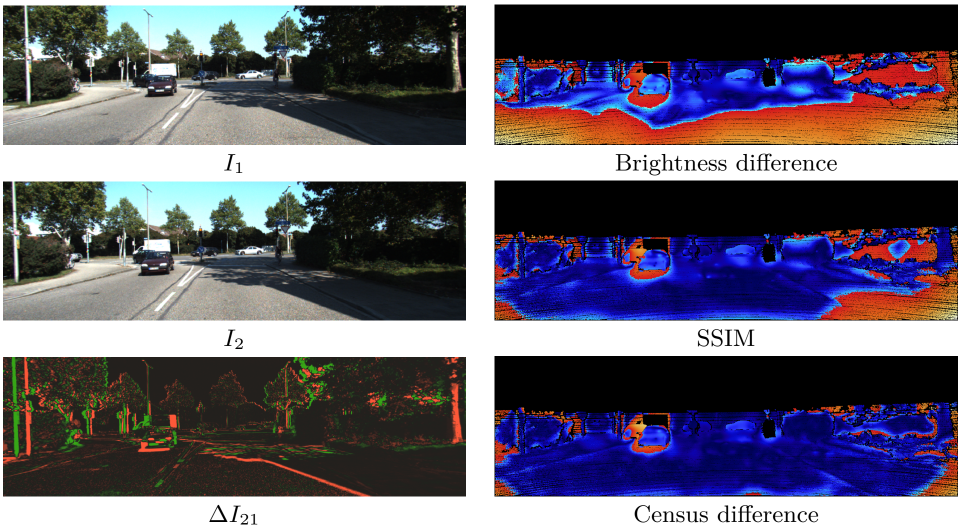


Figure 5.1 Photometric difference measures comparison. Left: A sample pair of images I_1, I_2 from the KITTI dataset and their difference ΔI_{21}. Right: Optical flow error maps for the tested photometric difference measures. Inliers in blue tones, outliers in red-yellow tones.

5.1.2 Occlusion handling

The next section evaluates the experiments with occlusion handling. We examine both occlusion detection by forward-backward optical flow consistency and three-frame occlusion reasoning. Table 5.2 shows the optical flow error for both methods. The experiments from the previous section (Section 5.1.1) serve as the baseline since they share all settings except occlusion handling.

Forward-backward consistency occlusion detection

First, the experiments with occlusion handling by forward-backward consistency constraint are evaluated. Table 5.3 shows the occlusion detection accuracy with this method.

Rows 1-4 in Table 5.2 show that with brightness difference as f_D, occlusion detection and masking leads to a decreased error on all datasets. The tested settings of occlusion penalty λ_O perform very similarly. More significant error decrease is observed on occluded pixels, which are less included in the data term and thus less likely to follow an incorrect photo-consistency objective.

As rows 7-9 (Table 5.2) show, the improvement to no occlusion masking is not as clear with Census difference f_D. We observe a small error increase with Sintel Clean, a small decrease on Sintel Final, and similar error fluctuations on KITTI.

Rows 13-15 (Table 5.2) show the experiment, where the occlusion reasoning is introduced in fine-tuning i.e., after the model is already trained on the given datasets and produces more precise optical flow estimates. Contrary to our expectations, this approach does not lead to better results - we observe higher or similar error on both Sintel and KITTI datasets compared to the baseline (i.e., fine-tuning with no occlusion reasoning).

Overall, we observe that forward-backward consistency occlusion detection does not lead to a significantly decreased optical flow error. The reason most probably lies in the occlusion detection accuracy. As we observe in Table 5.3, the precision reaches maximally about 0.45 and recall 0.5, meaning that more than a half of masked pixels in the data term in fact have a matching pixel in the second frame. On the other hand, the data term is still active for about half of occluded pixels. This corrupts the training - optical flow training on some non-occluded pixels is driven just by smoothness term, which potentially explains why we observe the increased error on non-occluded pixels of Sintel Clean.

Occlusion detection is also tested by checking the forward-backward consistency of optical flow produced by a supervised model. Table 5.3 shows an increase in precision and recall with respect to the unsupervised models. Nevertheless, the accuracy is still low, suggesting that the proposed detection method is not robust enough to detect occlusions reliably on the given optical flow estimates.

Three-frame occlusion handling

Next, the experiments with the three-frame occlusion reasoning are reviewed. Table 5.2 shows the optical flow estimation accuracy, Table 5.4 lists the occlusion detection accuracy measured as described in Section 4.2.2.

As it is shown in Table 5.2, the tested occlusion handling scheme leads to worse results in all cases. In some experiments, error on occluded pixels is lower than the respective baseline. However, the error on non-occluded pixels rises in all cases making the overall error always higher.

The potential cause can be observed in Table 5.4 that shows the accuracy of occlusion detection. The AUC metric is below 0.26 in all cases, meaning that is is very low. The problem is demonstrated on Figure 5.2. The estimated occlusion mask with $\lambda_P = 0$ is very noisy, observe e.g., the hand or dragon's

					Sintel AEPE [px]						KITTI	
					Clean			Final			Fl-all [%]	
Tr.	**Met.**	f_D	λ_O	λ_P	**ALL**	**NOC**	**OCC**	**ALL**	**NOC**	**OCC**	**2015**	**2012**
-	No	B	-	-	5.34	3.38	31.65	6.41	4.35	33.99	41.94	29.74
M	FwBw	B	8	-	5.35	3.58	29.00	6.16	4.28	31.31	40.33	26.42
M	FwBw	B	35	-	5.26	3.49	28.99	6.13	4.25	31.27	40.46	26.75
M	FwBw	B	70	-	5.23	3.47	28.82	6.12	4.24	31.21	40.50	26.61
M	3 Fr.	B	-	0	6.04	4.36	28.54	6.49	4.79	29.22	48.02	31.87
M	3 Fr.	B	-	5	6.20	4.49	29.09	7.30	5.48	31.70	44.31	28.22
-	No	C	-	-	4.08	2.31	27.74	5.24	3.36	30.54	26.35	13.25
M	FwBw	C	8	-	4.39	2.71	26.95	5.22	3.46	28.87	26.00	12.47
M	FwBw	C	70	-	4.26	2.65	25.84	5.08	3.37	28.02	26.40	12.62
M	3 Fr.	C	-	0	5.19	3.42	28.83	5.66	3.85	29.92	36.98	20.99
M	3 Fr.	C	-	5	4.93	3.17	28.55	5.64	3.83	29.88	34.72	19.02
M	3 Fr.	C	-	50	4.76	2.95	28.97	5.66	3.70	31.86	30.35	16.33
-	No	C	-	-	4.06	2.30	27.57	5.22	3.35	30.33	25.71	12.91
FT	FwBw	C	8	-	4.31	2.58	27.44	5.24	3.45	29.14	26.29	12.67
FT	FwBw	C	70	-	4.27	2.56	27.25	5.20	3.43	28.95	26.27	12.67
FT	3 Fr.	C	-	0	4.54	2.85	27.18	5.52	3.68	30.26	33.50	18.63
FT	3 Fr.	C	-	50	4.34	2.60	27.65	5.51	3.60	31.02	29.03	15.36

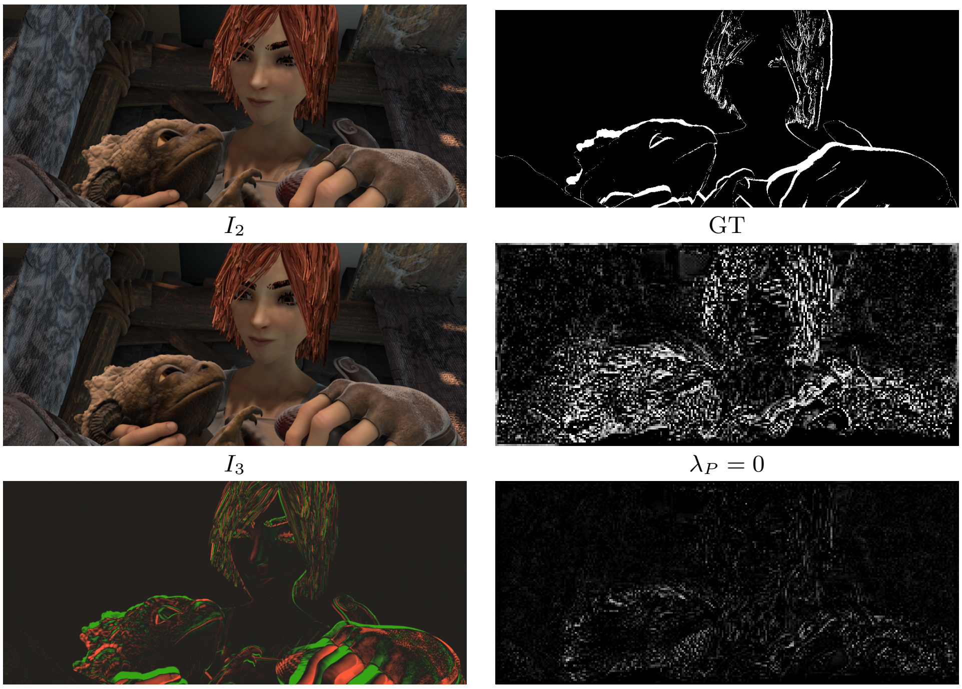
Table 5.2 Optical flow error with occlusion handling. Optical flow accuracy on Sintel and KITTI datasets. Column "Tr." designates when the occlusion handling was activated - during main training (M), later in fine-tuning (FT) or never (-). "Met." refers to the occlusion handling method - either no handling (No), forward-backward consistency check (FwBw) or three-frame reasoning (3 Fr.). Tested photometric difference measures f_D are brightness difference (B) and Census difference measure (C). Occlusion penalty λ_O (for fw-bw method) resp. occlusion prior λ_P (for three frame method) is a method-specific setting. Thin line underlines a baseline experiment for section below a thick line.

			Sintel occlusions					
			Clean			Final		
Training	f_D	λ_O	**P**	**R**	**F1**	**P**	**R**	**F1**
-	B	-	0.34	0.51	0.40	0.30	0.51	0.38
M	B	8	0.34	0.44	0.38	0.34	0.44	0.39
M	B	35	0.36	0.46	0.40	0.35	0.46	0.40
M	B	70	0.37	0.44	0.40	0.34	0.45	0.39
-	C	-	0.45	0.50	0.47	0.40	0.52	0.45
M	C	8	0.44	0.45	0.45	0.41	0.45	0.43
M	C	70	0.44	0.46	0.45	0.41	0.46	0.43
-	C	-	0.46	0.51	0.48	0.40	0.51	0.45
FT	C	8	0.45	0.44	0.44	0.42	0.45	0.43
FT	C	70	0.45	0.44	0.44	0.42	0.45	0.43
Supervised - Sintel			0.56	0.66	0.61	0.47	0.67	0.55

Table 5.3 Forward-backward consistency occlusion detection error. Precision (P), recall (R) and F1 score (F1) for occlusion detection on Sintel. Column "Training" designates when the occlusion handling was activated - during main training (M), later in fine-tuning (FT) or never (-). Tested photometric difference measures f_D are brightness difference (B) and Census difference measure (C). λ_O refers to the occlusion penalty. Results for a model trained with supervision (on Sintel) are listed for comparison.

face, but the real occlusions are undetected. With higher prior $\lambda_P = 50$, the noise gets lower, but the actual occluded areas are still mostly not detected.

This observation suggests that the mask M^l is rather selecting pixels from either warped I_1 or warped I_3 that happen to be photo-metrically consistent with the pixels in I_2 due to the estimated optical flow fields. When $\mathbf{f}^l_{2\to 1}$ or $\mathbf{f}^l_{2\to 3}$ is incorrect in some pixel, M^l tends to mask the error out and use only one, correct, direction of flow to fill the pixel in $\tilde{I}^l_2$.


I_2

GT

I_3

$\lambda_P = 0$

ΔI_{32}

$\lambda_P = 50$

Figure 5.2 Three-frame occlusion handling. Left: I_2 and I_3 of a sample triplet from Sintel (clean) and their difference ΔI_{32}. Right: forward occlusion (i.e. from I_2 to I_3) ground truth and estimates for different mask prior λ_P setting. White denotes pixels occluded in I_3.

			Sintel occlusions							
			Clean				Final			
				Thr. 0.75				Thr. 0.75		
Training	f_D	λ_P	**AUC**	**P**	**R**	**F1**	**AUC**	**P**	**R**	**F1**
M	B	0	0.22	0.20	0.45	0.27	0.21	0.21	0.43	0.28
M	B	5	0.25	0.34	0.37	0.35	0.23	0.33	0.35	0.34
M	C	0	0.16	0.16	0.44	0.23	0.17	0.17	0.41	0.24
M	C	5	0.20	0.22	0.39	0.28	0.21	0.24	0.37	0.29
M	C	50	0.17	0.42	0.03	0.05	0.14	0.38	0.02	0.04
FT	C	0	0.23	0.28	0.42	0.34	0.23	0.25	0.40	0.31
FT	C	50	0.17	0.52	0.01	0.02	0.15	0.49	0.01	0.02

Table 5.4 Three-frame occlusion detection error. Occlusion detection accuracy on Sintel - Area under curve (AUC) and precision, recall and F1 score with the given threshold. Column “Training” denotes when the training with occlusion reasoning is activated - during main training (M) or fine-tuning (FT). f_D denotes the photometric difference measure (B - brightness difference, C - Census difference measure).

5.1.3 Forward-backward consistency term

Next, the experiments with forward-backward consistency term (see Section 4.2.3) are reviewed. Table 5.5 shows the optical flow accuracy on Sintel and KITTI. Results from the experiments with photometric difference measures (Section 5.1.1) serve as a baseline since they are performed with the same settings except for the activated consistency term L_C (Eq. 3.20).

As the table shows, $\lambda_C = 3$ is a too high setting that increases the error (except for KITTI 2012). For other settings, the error on the Sintel dataset is slightly decreased except Sintel clean pass with Census photometric difference, where a small increase is observed. However, a dramatic error decrease is observed on the KITTI dataset. Thus, we conclude that the consistency term is beneficial in our setting.

		Sintel AEPE [px]						KITTI	
			Clean			Final		Fl-all [%]	
f_D	λ_C	ALL	NOC	OCC	ALL	NOC	OCC	2015	2012
B	0	5.34	3.38	31.65	6.41	4.35	33.99	41.94	29.74
B	0.3	5.23	3.40	29.71	6.18	4.21	32.53	39.62	26.15
C	0	4.08	2.31	27.74	5.24	3.36	30.54	26.35	13.25
C	0.3	4.22	2.56	26.42	5.19	3.38	29.40	25.14	12.50
C	0.5	4.17	2.48	26.78	5.18	3.38	29.35	25.68	12.73
C	3	4.68	2.92	28.27	5.27	3.46	29.42	26.87	12.78

Table 5.5 Forward-backward consistency term. Performance on Sintel and KITTI datasets for different weights of the consistency term λ_C. f_D denotes photometric difference measure (B - brightness difference, C - Census difference measure).

5.1.4 Dataset size

Table 5.6 lists the results from experiments with the training dataset size. Figure 5.3 shows qualitative assessment on a sample picked from the KITTI dataset.

The table shows that errors on Sintel are slightly higher with both big and small training datasets compared to the baseline. However, on KITTI, the differences get stronger, with both experiments lagging behind the baseline by a significant amount.

Contrary to the expectations, training on a large amount of data does not increase the accuracy. This might have two potential reasons. First, despite our efforts, the added data may not be appropriate for training. Either the average amount of movement between the added frames lacks behind the standard datasets, and thus the training becomes less challenging. Alternatively, the scene cut detection failed in too many cases causing a failure in training.

The second option is that the unsupervised objective does not require many training samples, and the standard training dataset already saturates

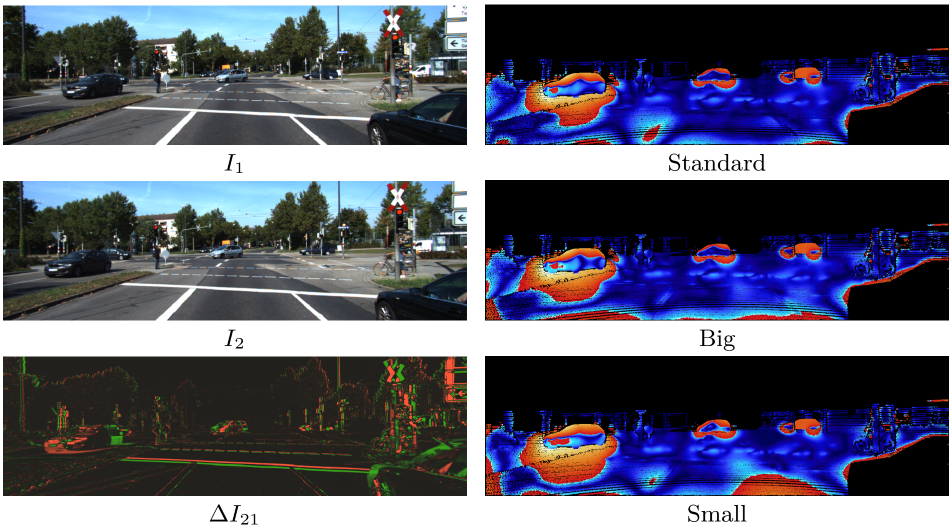
	Sintel AEPE [px]						KITTI	
	Clean			Final			Fl-all [%]	
Dataset	**ALL**	**NOC**	**OCC**	**ALL**	**NOC**	**OCC**	**2015**	**2012**
Standard	5.34	3.38	31.65	6.41	4.35	33.99	41.94	29.74
Big	5.34	3.37	31.74	6.74	4.54	36.26	43.62	31.43
Small	5.79	3.75	33.21	6.91	4.70	36.56	46.43	34.31

Table 5.6 Dataset size performance. Performance on Sintel and KITTI datasets for different training dataset sizes. The "Standard" is composed of KITTI and Sintel samples (ca. 15K), "Big" collects wide range of scenes - see Section 4.2.4 (ca. 221K), "Small" selects only a few samples from Sintel and KITTI (80 pairs).

the capacity. Indeed, the objective is more universal - it formulates the optical flow estimation as an optimization of photoconsistency combined with smoothness. We hypothesize that the network does not need many different examples to "learn" to follow this objective.

The results from the experiment with a small training dataset partially back the second option. Despite the clear performance gap, the difference is not proportional to the dataset reduction - 80 pairs vs. 15000 pairs. Even the qualitative assessment (Figure 5.3) shows that although the number of outliers is higher, the results are not substantially different.

To conclude, we observe that the ability to use vast amounts of training data with unsupervised loss does not necessarily lead to a higher optical flow accuracy. On the contrary, even small training dataset can lead to comparable results. We hypothesize that this effect is caused by the unsupervised formulation of the loss function.



I_1

Standard

I_2

Big

ΔI_{21}

Small

Figure 5.3 Dataset size experiments. Left: KITTI sample frames I_1 and I_2 and their difference ΔI_{21}. Right: Optical flow error maps for models trained on different dataset sizes. Inliers in blue tones, outliers in red-yellow colors. The "Standard" dataset is composed of KITTI and Sintel samples (ca. 15K), "Big" collects wide range of scenes - see Section 4.2.4 (ca. 221K), "Small" selects only a few samples from Sintel and KITTI (80 pairs).

5.2 Semi-supervised training on a single/close domain

In this section, we present and analyze the results of experiments with semi-supervised fine-tuning and domain adaptation (Section 4.3). The goal of this series of experiments is to start with a model trained in a supervised manner on Sintel and improve its accuracy on either Sintel or KITTI. This is attempted by introducing the unsupervised samples to the training with the proposed semi-supervision method. We either try to fine-tune the model for the Sintel dataset or adapt it for KITTI.

The results from the experiments are listed in Table 5.7. The table includes results of the initial supervised model and also adds results from an unsupervised experiment (Photometric difference measures - census transform, see Section 5.1.1) for comparison.

5.2.1 Semi-supervised fine-tuning

Semi-supervised fine-tuning attempts to increase the performance on Sintel by adding unsupervised samples from the Sintel movie.

Rows 3 and 4 of the Table 5.7 show that semi-supervised fine-tuning fails to improve the performance of the supervised model on Sintel (row 2). On the contrary, the error becomes more significant. The first two lines of the table comparing supervised and unsupervised training might indicate the reason. Since unsupervised training is not able to bring the error on Sintel lower than supervised training, it is hard to expect that adding it to supervised training will yield a more accurate optical flow estimation.

5.2.2 Semi-supervised domain adaptation

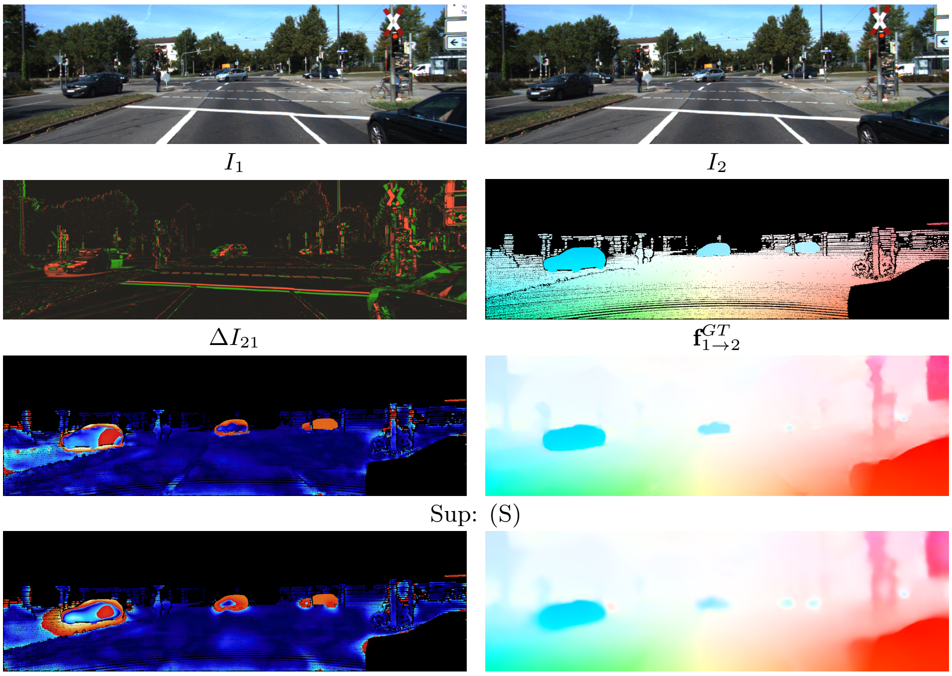
Semi-supervised domain adaptation for KITTI also gives unsuccessful results as rows 5 and 6 (Table 5.7) show. By adding unsupervised KITTI raw samples to the training, the error on KITTI significantly rises. Figure 5.4 shows the comparison of supervised and semi-supervised training on KITTI. We see that with semi-supervision, the optical flow becomes much smoother and stops respecting the edges of moving objects.

This result gives us the following insight. Estimating optical flow with the supervised objective and unsupervised objective are two different tasks from the point of the network. By presenting Sintel samples with the supervised loss and KITTI samples with the unsupervised loss, the network learns (to a certain degree) to solve the two domains with the respective objectives. This may explain why with domain adaptation, only a small error rise is observed on Sintel, whereas the accuracy on KITTI deteriorates significantly.

The models start to use the unsupervised objective on KITTI; however, the training is not long enough to achieve the performance of the unsupervised method.

		Sintel AEPE [px]						KITTI	
		Clean			Final			Fl-all [%]	
Experiment	**TS**	**ALL**	**NOC**	**OCC**	**ALL**	**NOC**	**OCC**	**2015**	**2012**
Unsup: (K+S)	-	4.08	2.31	27.74	5.24	3.36	30.54	26.35	13.25
Sup: (S)	-	2.16	1.19	15.15	2.74	1.76	15.89	25.55	12.38
Semi: (S→Sm)	S	2.32	1.26	16.42	2.92	1.86	17.21	26.41	12.79
Semi: (S→Sm)	L	2.30	1.25	16.41	3.11	1.95	18.56	27.02	12.77
Semi: (S→Kr)	S	2.24	1.23	15.74	2.93	1.87	17.10	29.84	15.69
Semi: (S→Kr)	L	2.14	1.16	15.29	2.89	1.84	16.97	29.51	15.27

Table 5.7 Semi-supervised training on single/close domain. Performance on Sintel and KITTI datasets for experiments with the proposed semi-supervision method. Column "Experiment" refers to the training method (supervised/unsupervised/semi-supervised) and the training datasets in brackets - S: Sintel, Sm: Sintel movie, K: KITTI, Kr: KITTI raw. "TS" refers to training scheme - S: short or L: long (see experiments - Section 4.3). Arrow "→" separates supervised and unsupervised datasets by semi-supervision. The unsupervised experiment (Photometric difference measures - census transform, see Section 5.1.1) is listed for a comparison.



I_1

I_2

ΔI_{21}

$\mathbf{f}^{GT}_{1\to 2}$

Sup: (S)

Semi: (S→Kr), TS: S

Figure 5.4 Semi-supervised training on single/close domain. First row: KITTI sample frames I_1 and I_2. Second row: Frames difference ΔI_{21} and ground-truth (sparse) optical flow $\mathbf{f}^{GT}_{1\to 2}$. Next rows: Endpoint-error map - inliers in blue tones, outliers in red-yellow colors (left) and optical flow estimates (right) for supervised and semi-supervised models. The abbreviation Sup/Semi refers to the training method supervised resp. semi-supervised with the training datasets in brackets - S: Sintel, Kr: KITTI raw. Arrow "→" separates supervised and unsupervised datasets by semi-supervision. "TS: S" refers to the short training scheme

5.3 Semi-supervised training on a distant domain

Next, we focus on experiments with semi-supervised training for a distant domain adaptation (Section 4.4). The goal is to make the estimation network combine supervised and unsupervised objectives so that it performs close to the unsupervised methods on data from a distant domain while maintaining the performance on the labeled domain.

The main results are listed in Table 5.8. Results of parameter search experiments for constrained semi-supervision are in Table 5.10, highlighted lines are selected for the general comparison in the main results table.

Let us first discuss the general results and then focus on constrained semi-supervision parameter settings.

5.3.1 General observations

In this section, we compare the performance of classical methods, supervised, unsupervised, and semi-supervised models, as Table 5.8 shows. The main focus is on the decrease of error on CF+ and the maintenance of performance on the Sintel dataset. Note that with CF+, it is more meaningful to observe performance on the foreground areas since optical flow on the background is often not well defined.

Classical methods

Rows 1-3 of Table 5.8 show that Horn-Schunck and other classical methods even with no fine-tuning generalize better on the CF+ dataset compared to the Sintel-supervised neural networks. However, when considering Sintel accuracy, they stay far behind. This gives us the motivation to achieve the performance of classical methods on CF+ while at the same time be more accurate on Sintel.

Supervised methods

Rows 4 and 5 of Table 5.8 (*Sup: (C,T)* and *Sup: (C,T,S)*) list two models made available by PWC-Net authors [33] that are trained on FlyingChairs and FlyingThings3D [25] datasets resp. also including Sintel. Row 6 (*Sup: (C,T,S) - cont.*) shows a model that starts with weights of *Sup: (C,T,S)* and the training is continued on Sintel using supervised loss.

Overall, we observe that the supervised methods trained on Sintel or FlyingThings3D fail on the CF+ dataset (Table 5.8, rows 4-6). Figure 5.5 (see row 5) indicates abruptly outlying estimates on constant intensity regions. Problems also occur on object texture changes. This finding is in line with [42].

Interestingly, row 6 shows that continued supervised training seems to decrease the error on CF+. It is most likely caused by differences in training to the original model, probably because we skip additive white noise augmentation in our setting. This shows that the distant domain transfer ability of supervised methods to CF+ is sensitive to the precise setting of training.

Unsupervised models

Table 5.8, rows 7 and 8 show the performance of unsupervised methods on CF+. These are picked for comparison from Section 5.1.3 - *Forward-backward consistency term* because the unsupervised loss in semi-supervised experiments uses the same setting.

The test errors on Sintel and KITTI dataset stay far behind the supervised methods. However, they do not suffer from distant domain transfer issues as the supervised models - the performance on the CF+ dataset is significantly better. Figure 5.5 (row 6) shows that the estimated flow field is smoother, with no abrupt outliers.

We hypothesize that although the unsupervised objective is unable to properly handle the effects of occlusions, motion blur, local ambiguities, etc., yet, it is more universal than a supervised objective on a single domain. Therefore, we expect it to perform better on a distant domain.

Constrained semi-supervision on a single domain

With semi-supervision combining Sintel and Sintel movie (Table 5.8, row 10), the test error on CF+ significantly drops while the error on Sintel changes just slightly. We attribute this result partially to the better-performing supervised training. However, semi-supervision leads to a more significant decrease, suggesting that adding the unsupervised loss with the proposed method makes the model perform closer to unsupervised methods with only minor changes on the Sintel domain.

Constrained semi-supervision including the distant domain

When the semi-supervised model is explicitly presented with the samples from CF+, the error on this distant domain drops significantly to the level of the unsupervised methods (see row 11 of Table 5.8). Note that the error is also significantly below the semi-supervision on a single domain. Again, the error on Sintel stays virtually the same.

We hypothesize that since the images from the other domain are presented, the network starts to recognize it and optimize the unsupervised criterion specifically on these samples. However, the supervised constraint prevented to apply the same criterion on the supervised samples.

Unconstrained semi-supervision

The results of all experiments with unconstrained semi-supervision are listed separately in Table 5.9. The performance on CF+ is similar for all settings, especially on the foreground regions. With low λ_U, the accuracy on Sintel and KITTI are maintained with respect to the initial model; however, a significant error rise is observed with higher λ_U.

Table 5.8 (row 9) shows the comparison of $\lambda_U = 0.1$ experiment with constrained semi-supervision. Error on Sintel is similar, but the improvement in CF+ test error is not as significant as with the proposed constraining.

The observations correspond to the expectations - with small unsupervised term weight, the training is not able to introduce the unsupervised objective to the model. When we attempt to promote it more with higher λ_U, the accuracy on the supervised domain is lost. Also note the over-fitting behavior during the training as described in the experiments section (Sec. 4.4.3) that was not observed with the constraining.

Overall, this experiment suggests that the proposed gradient constraining method combines the supervised and unsupervised training more effectively than naive semi-supervision.

Supervised CF training

The model supervised on CF+ was able to improve on the dataset while maintaining the performance on Sintel (see Table 5.8, last row). Evaluated on whole frames, it does not surpass constrained semi-supervision. However, as it was already mentioned, the background flow is often not well defined; thus, this metric is not as relevant.

The performance margin to a constrained semi-supervision on the foreground areas is not as large as, e.g., the margin between supervised and unsupervised methods on Sintel, suggesting that CF+ features complicated scenes that are hard to solve even with supervision.

Error increase on KITTI

We observe that the test error on KITTI rises for both semi-supervised and supervised experiments by some amount. Again, this effect seems to be mostly connected to our modified supervision method. In the experiments, we also observe that it depends on the number of training iterations.

	CF+ AEPE [px]			Sintel AEPE		KITTI 2015
		median		[px]		[%]
Experiment	ALL	ALL	FG	Clean	Final	Fl-all
Horn-Schunck [1]	8.34	3.49	12.17	8.73*	9.61*	–
Classic+NLfast [9]	13.35	7.05	9.27	9.12*	10.08*	–
Brox2011 [7]	9.05	3.27	8.28	7.56*	9.11*	–
Sup: (C,T)	66.97	41.88	22.77	2.44	3.82	34.26
Sup: (C,T,S)	74.23	33.54	18.21	1.78	2.41	10.56
Sup: (C,T,S) - cont.	30.44	14.73	11.30	1.69	2.22	14.73
Unsup [Br]: (C,K+S)	10.60	4.80	7.99	5.23	6.18	39.62
Unsup [Ce]: (C,K+S)	15.06	9.05	8.65	4.22	5.19	25.14
Unconstr. semi: (S)	25.76	15.19	10.63	1.79	2.19	12.22
Semi: (S→Sm)	17.36	8.41	8.91	1.81	2.49	16.88
Semi: (S→CF)	7.88	3.79	6.65	1.79	2.25	18.89
Sup: (C,T,S,S+CF)	8.19	3.54	5.62	1.81	2.24	17.37

Table 5.8 Semi-supervision on a distant domain - main results. Performance on Creative Flow+, Sintel and KITTI datasets. "Experiment" denotes the training setup - supervised (Sup), unsupervised (Unsup, [f_D]) or semi-supervised (Semi). Training datasets in brackets - C: FlyingChairs, T: FlyingThings3D [25], S: Sintel, Sm: Sintel movie, K: KITTI, CF: Creative Flow+. Arrow "→" separates supervised and unsupervised datasets. All numbers except columns marked median and Fl-all, are mean endpoint errors over all test samples. Median is computed across individual sample average EPEs. For classical methods (first 3 rows), we list the results from [42]. Results marked with a star (*) come from the official test benchmark instead of own train/test split.

5.3.2 Parameter selection discussion

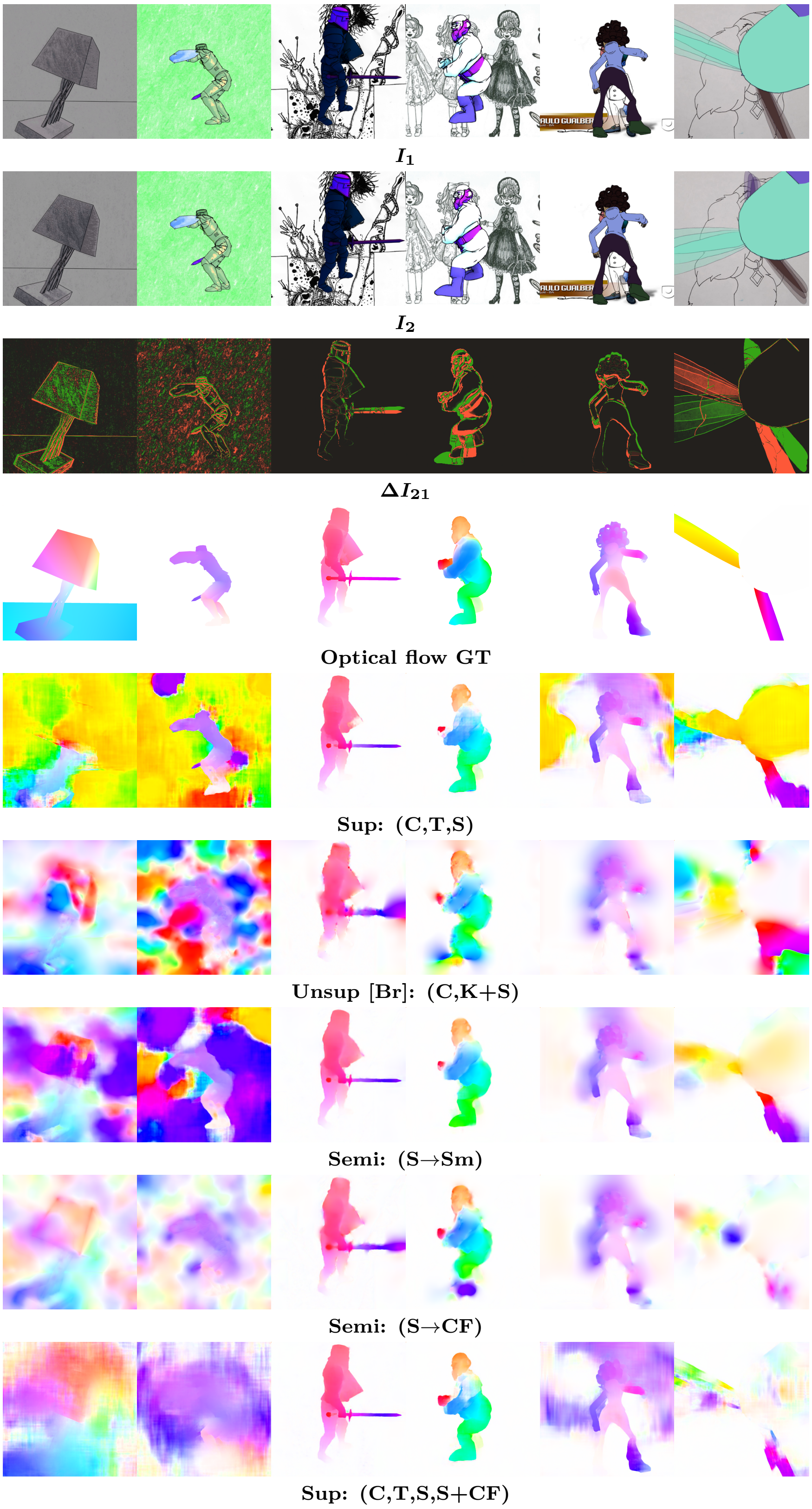
Let us first discuss the results the experiments with various parameter setting (Table 5.10). We make the following observations.

Robust supervised loss. (row 2) Although it leads to a significant error decrease on CF+ compared to the other experiments, error on the Sintel dataset rises. We hypothesize that this behavior is linked to constraining during semi-supervision - as robust supervised loss decreases the influence of outliers, it allows for more substantial changes in network parameters towards unsupervised criterion compared to L2 loss. This, however, also leads to performance loss on Sintel.

Census photometric difference (rows 5 and 9) seems to behave slightly worse than the brightness difference on CF+ with no significant differences on the Sintel dataset.

Learning rate setting of 1e-7 seems to restrict the training too much since the error on CF+ stays high. The other values 1e-5 and 1e-6 lead to comparable results on Sintel. On CF+, the higher setting in combination with the long training schedule seems to perform better, but on the other hand, it also allows for a more significant rise of the error on KITTI.

Based on the observations, the experiments highlighted in Table 5.10 are selected for the main comparison.



$\boldsymbol{I_1}$

$\boldsymbol{I_2}$

$\boldsymbol{\Delta I_{21}}$

Optical flow GT

Sup: (C,T,S)

Unsup [Br]: (C,K+S)

Semi: (S→Sm)

Semi: (S→CF)

Sup: (C,T,S,S+CF)

Figure 5.5 Semi-supervision on a distant domain - qualitative assessment. Input images (first two rows) with a color coded difference visualization (third row); the ground truth flow and flow estimates for selected methods (following rows).

	CF+		**Sintel**		**KITTI**	
	AEPE [px]		**AEPE** [px]		**Fl-all** [%]	
λ_U	**All**	**Fg**	**Clean**	**Final**	**2015**	**2012**
0.1	**25.76**	**10.63**	**1.79**	**2.19**	**12.22**	**7.30**
1	24.91	9.95	2.54	3.10	22.02	11.50
2	24.07	10.12	2.95	3.63	29.25	17.32

Table 5.9 Unconstrained semi-supervision. Performance on CF+, Sintel and KITTI datasets for experiments with unconstrained (naive) semi-supervision. λ_U is the unsupervised loss weight. As explained in Section 4.4.3, the experiments are evaluated before error rise on CF+. Bold typeface marks the experiment selected for the final evaluation.

					CF+		**Sintel**		**KITTI**	
					AEPE [px]		**AEPE** [px]		**Fl-all** [%]	
Experiment	**TS**	f_D	L_{sup}	**LR**	**All**	**Fg**	**Clean**	**Final**	**2015**	**2012**
Semi: (S→Sm)	S	B	L2	1e-6	19.97	9.04	1.86	2.39	13.05	7.80
		B	R	1e-6	14.47	8.18	2.07	2.78	15.88	8.55
		B	L2	1e-5	20.81	9.67	1.84	2.40	15.65	8.68
		B	L2	1e-7	30.74	11.38	1.83	2.31	13.15	7.66
		C	L2	1e-6	22.20	9.32	1.83	2.35	12.86	7.60
	L	B	L2	1e-6	20.24	9.27	1.85	2.42	13.61	7.93
		B	**L2**	**1e-5**	**17.36**	**8.91**	**1.81**	**2.49**	**16.88**	**8.98**
		B	L2	1e-7	30.98	11.38	1.82	2.32	12.76	7.54
		C	L2	1e-6	19.35	9.23	1.83	2.32	12.97	7.71
Semi: (S→CF)	S	B	L2	1e-6	9.59	7.18	1.84	2.26	15.26	8.92
	L	B	L2	1e-6	9.14	7.07	1.83	2.24	15.46	9.09
		B	**L2**	**1e-5**	**7.88**	**6.65**	**1.79**	**2.25**	**18.89**	**10.89**

Table 5.10 Semi-supervision on a distant domain - parameter comparison. Performance on CF+ (whole images/foreground), Sintel and KITTI datasets for experiments with semi-supervision. Column “Experiment” refers to the dataset combination - S: Sintel, Sm: Sintel movie, CF: Creative Flow+. Arrow “→” separates supervised and unsupervised datasets. “TS” refers to training scheme - S: short or L: long (see experiments - Section 4.4.1). f_D marks the type of photometric difference function - B: brightness, C: census. L_{sup} is the supervised error L2 or R: robust. LR refers to the initial learning rate. Bold typeface marks experiments selected for the final evaluation.

Chapter 6

Conclusions

The thesis has two main contributions. First, a number of current techniques in unsupervised optical flow training are analyzed, and a direct comparison is established. Second, a new semi-supervision method based on constrained gradient descent is proposed, and its potential benefits are demonstrated.

In the unsupervised training analysis, we make the following conclusions.

- Brightness difference, census transform, and structural similarity are tested as photometric difference measures as the results are compared. We show that the census transform consistently leads to more accurate results on all tested datasets.

- Forward-backward consistency occlusion detection is implemented and tested in both main training and fine-tuning. In the experiments, it does not lead to a significant improvement. Poor detection accuracy is presumed to be the main culprit.

- Three-frame occlusion reasoning is also integrated and tested. In both main training and fine-tuning it leads to an error increase compared to the respective baselines. The observations suggest that the training process is corrupted because instead of occlusions, optical flow estimation errors are being detected.

- The effect of forward-backward consistency loss term is tested. Experiments suggest it can contribute to more accurate optical flow estimates with specific weight settings.

- Training dataset size influence is analyzed. Curiously, the experiments show that a large amount of training data does not necessarily lead to a performance increase. Even more interestingly, training on an extra small number of samples does not lead to a catastrophic loss of accuracy. This suggests that the popular rule from the supervised training: “CNN training needs a large amount of data” might not fully apply to unsupervised training.

Contributions regarding semi-supervised training are the following.

- A novel method combining supervised and unsupervised objectives is presented. The training is formulated as constrained gradient descent on a loss function that includes terms from unsupervised training.
- The method is tested on supervised and unsupervised objectives in a single domain of Sintel i.e. for semi-supervised fine-tuning combining Sintel dataset and Sintel movie unlabeled samples. No accuracy improvement in optical flow estimation was observed in the experiment.
- Semi-supervised domain adaptation for a close unlabeled domain using the proposed method is tested. Specifically, Sintel fine-tuned model is presented with unlabeled KITTI raw frames. This approach does not lead an accuracy increase on KITTI.
- Semi-supervision is tested for an adaptation to an unlabeled distant domain, where supervised training leads to abrupt estimates. Sintel fine-tuned model is adapted to Creative Flow+ domain. Experiments show that the proposed semi-supervised training helps to improve results on Creative Flow+ significantly. At the same time, the model performance on Sintel does not drop. Moreover, this effect is even observed without using any samples from Creative Flow+. Upon introducing the images from the distant domain (with no GT), we are able to bring the error on the distant domain even lower.

The points of the assignment were elaborated as follows.

- A detailed analysis of state-of-the-art self-supervised optical flow training methods was performed. The major methods were described in Section 2.
- PWC-Net [33] was selected as a recent well-performing optical flow estimation method. The selected PWC-Net neural network architecture is described in more detail in Section 4.1.1 and is used throughout the whole work.
- A new method for semi-supervised training (i.e., partly self-supervised) is proposed in Section 3.4.2. The unsupervised objective in this method uses the conclusions of a preceding detailed analysis of techniques in unsupervised training (Section 4.2).
- The method is evaluated in different scenarios on suitable datasets (Section 4) and the results are discussed in Section 5.

Bibliography

[1] B. K. Horn and B. G. Schunck, "Determining optical flow", *Artificial intelligence*, Artificial intelligence, vol. 17, no. 1, pp. 185–203, 1981.

[2] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision", in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'81, San Francisco, CA, USA, 1981, pp. 674–679.

[3] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence", in *Computer Vision — ECCV '94*, J.-O. Eklundh, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 151–158.

[4] F. Stein, "Efficient computation of optical flow using the census transform", in *Pattern Recognition*, C. E. Rasmussen, H. H. Bülthoff, B. Schölkopf, and M. A. Giese, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 79–86.

[5] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity", *IEEE Transactions on Image Processing*, vol. 13, p. 600, Apr. 2004.

[6] W. Trobin, T. Pock, D. Cremers, and H. Bischof, "An unbiased second-order prior for high-accuracy motion estimation", in *DAGM-Symposium*, 2008.

[7] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation", *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 3, pp. 500–513, 2010.

[8] C. Levy (Director), *Sintel*, Blender Institute, 2010.

[9] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles", in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2010, pp. 2432–2439.

[10] N. Sundaram, T. Brox, and K. Keutzer, “Dense point trajectories by GPU-accelerated large displacement optical flow”, in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 438–451.

[11] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, “A naturalistic open source movie for optical flow evaluation”, in *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 611–625.

[12] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the KITTI vision benchmark suite”, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[13] I. Hubert (Director), *Tears of steel*, Blender Institute, 2012.

[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset”, *International Journal of Robotics Research (IJRR)*, 2013.

[15] P. Vazquez (Director), *Caminandes 1: Llama drama*, Blender Institute, 2013.

[16] ——, *Caminandes 2: Gran dillama*, Blender Institute, 2013.

[17] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization”, *arXiv preprint arXiv:1412.6980*, 2014.

[18] D. Sun, S. Roth, and M. J. Black, “A quantitative analysis of current practices in optical flow estimation and the principles behind them”, *International Journal of Computer Vision*, vol. 106, no. 2, pp. 115–137, Jan. 2014.

[19] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox, “FlowNet: Learning optical flow with convolutional networks”, in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification”, *IEEE International Conference on Computer Vision (ICCV 2015)*, vol. 1502, 2015.

[21] M. Auvray (Director), *Cosmos laundromat*, Blender Institute, 2015.

[22] M. Menze and A. Geiger, “Object scene flow for autonomous vehicles”, presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3061–3070.

[23] A. Ahmadi and I. Patras, “Unsupervised convolutional neural networks for motion estimation”, in *2016 IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, pp. 1629–1633.

[24] G. Long, L. Kneip, J. M. Alvarez, H. Li, X. Zhang, and Q. Yu, "Learning image matching by simply watching video", in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer International Publishing, 2016, pp. 434–450.

[25] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4040–4048.

[26] P. Vazquez (Director), *Caminandes 3: Llamigos*, Blender Institute, 2016.

[27] J. J. Yu, A. W. Harley, and K. G. Derpanis, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness", in *Computer Vision – ECCV 2016 Workshops*, G. Hua and H. Jégou, Eds., ser. Lecture Notes in Computer Science, Springer International Publishing, 2016, pp. 3–10.

[28] S. Alletto, D. Abati, S. Calderara, R. Cucchiara, and L. Rigazio, "TransFlow: Unsupervised motion flow by joint geometric and pixel-level estimation", *arXiv:1706.00322 [cs]*, Jun. 1, 2017. arXiv: `1706.00322`.

[29] W.-S. Lai, J.-B. Huang, and M.-H. Yang, "Semi-supervised learning for optical flow with generative adversarial networks", in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017, pp. 354–364.

[30] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha, "Unsupervised deep learning for optical flow estimation", in *Thirty-First AAAI Conference on Artificial Intelligence*, Feb. 12, 2017.

[31] J. Janai, F. Guney, A. Ranjan, M. Black, and A. Geiger, "Unsupervised learning of multi-frame optical flow with occlusions", presented at the Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 690–706.

[32] S. Meister, J. Hur, and S. Roth, "UnFlow: Unsupervised learning of optical flow with a bidirectional census loss", in *Thirty-Second AAAI Conference on Artificial Intelligence*, Apr. 27, 2018.

[33] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-net: CNNs for optical flow using pyramid, warping, and cost volume", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943.

[34] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu, "Occlusion aware unsupervised learning of optical flow", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4884–4893.

[35] X. Xiang, M. Zhai, R. Zhang, Y. Qiao, and A. El Saddik, "Deep optical flow supervised learning with prior assumptions", *IEEE Access*, vol. 6, pp. 43 222–43 232, 2018.

[36] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose", presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1983–1992.

[37] B. Castellano. (Nov. 26, 2019). GitHub: Breakthrough/PySceneDetect, [Online]. Available: `https://github.com/Breakthrough/PySceneDetect` (visited on 11/27/2019).

[38] H.-Y. Lai, Y.-H. Tsai, and W.-C. Chiu, "Bridging stereo matching and optical flow via spatiotemporal correspondence", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1890–1899.

[39] P. Liu, I. King, M. R. Lyu, and J. Xu, "DDFlow: Learning optical flow with unlabeled data distillation", *arXiv:1902.09145 [cs]*, Feb. 25, 2019. arXiv: `1902.09145`.

[40] P. Liu, M. Lyu, I. King, and J. Xu, "SelFlow: Self-supervised learning of optical flow", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4571–4580.

[41] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 240–12 249.

[42] M. Shugrina, Z. Liang, A. Kar, J. Li, A. Singh, K. Singh, and S. Fidler, "Creative flow+ dataset", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[43] Wikipedia contributors, *Optical flow — Wikipedia, The Free Encyclopedia*. 2019.

[44] M. Zhai, X. Xiang, R. Zhang, N. Lv, and A. El Saddik, "Learning optical flow using deep dilated residual networks", *IEEE Access*, vol. 7, pp. 22 566–22 578, 2019.

[45] J Utah. Driving downtown - chicago 4k - USA, [Online]. Available: `https://www.youtube.com/watch?v=kOMWAnxKq58` (visited on 11/13/2019).

[46] ——, London 4k - theater district - driving downtown UK, [Online]. Available: `https://www.youtube.com/watch?v=paVB7zNvbOE` (visited on 11/13/2019).

[47] ——, Mississippi coast 4k - deep south sunrise drive, [Online]. Available: `https://www.youtube.com/watch?v=8SVyB8gNxbg` (visited on 11/13/2019).

[48] ——, New orleans 4k - sunset drive - USA, [Online]. Available: https://www.youtube.com/watch?v=GZUEaZHd_uI (visited on 11/13/2019).

[49] ——, New york city 4k - wall street - driving downtown USA, [Online]. Available: https://www.youtube.com/watch?v=C8vdEnP8pH8 (visited on 11/13/2019).

[50] New Trailer Buzz. Top upcoming 2018 blockbuster movies trailer compilation, [Online]. Available: https://www.youtube.com/watch?v=DdeDeWiEgxU (visited on 11/12/2019).

[51] ——, Top upcoming animation movie trailers 2018 (part 2) | trailer compilation, [Online]. Available: https://www.youtube.com/watch?v=vHl4UcQDLDI (visited on 11/12/2019).

[52] ——, Upcoming horror film trailers 2018 | trailer compilation, [Online]. Available: https://www.youtube.com/watch?v=JAGt450pGmM (visited on 11/12/2019).

Appendix A

CD contents

The description of each directory on the CD is written in table A.1.

Directory name	**Description**
sources	Software source code
thesis	This thesis in pdf format

Table A.1 Table describing the contents of the root directory on the CD.